



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación :

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

Título del proyecto:

DESARROLLO DE UN SISTEMA MULTIJUGADOR EN RED
SOBRE UN JUEGO BASADO EN EL “SCRABBLE”

Alumno: Carolina Flores Ugarriza

Tutor: Eduardo Alfaro Larragueta

Pamplona, Febrero de 2010

ÍNDICE

ÍNDICE	1
Listado de Figuras	4
CAPÍTULO 1: INTRODUCCIÓN	7
1.1. INTRODUCCIÓN	8
1.2. OBJETIVOS	9
1.3. DESCRIPCIÓN DEL PROBLEMA	9
1.4. REQUISITOS	10
1.4.1. Requisitos generales	10
1.4.2. Requisitos del Interfaz	10
1.4.3. Requisitos de Hardware y de sistema	12
1.4.4. Requisitos de la base de datos	12
1.5. SOLUCIÓN PROPUESTA	12
1.5.1. Lenguaje de Programación	13
1.5.2. Base de Datos	14
CAPÍTULO 2: HERRAMIENTAS	15
2.1. NetBeans IDE	15
2.2. XAMPP	15
2.3. Photoshop	16
CAPÍTULO 3: ANÁLISIS DEL PROBLEMA	17
3.1. DIAGRAMA DE CASOS DE USO: General	17
3.1.1. Caso de Uso: Realizar Registro	19
3.1.2. Caso de Uso: Ver Rankings	19
3.1.3. Caso de Uso: Realizar Login	19
3.1.4. Caso de Uso: Desconectarse	19
3.1.5. Caso de Uso: Consultar Cuenta	20
3.1.6. Caso de Uso: Buscar Jugadores	20
3.1.7. Caso de Uso: Consultar Ayuda	20
3.1.8. Caso de Uso: Jugar	20
3.1.8.1. Caso de Uso: Identificarse	22
3.1.8.2. Caso de Uso: Id como Registrado	22
3.1.8.3. Caso de Uso: Id como Invitado	22
3.1.8.4. Caso de Uso: Entrar Salón	22
3.1.8.5. Caso de Uso: Entrar Sala	23
3.1.8.6. Caso de Uso: Entrar Mesa	23
3.1.8.7. Caso de Uso: Configurar Mesa	24
3.1.8.8. Caso de Uso: Abandonar Mesa	24
3.1.8.9. Caso de Uso: Salir Sala	24
3.1.8.10. Caso de Uso: Chatear	25
3.1.8.11. Caso de Uso: Salir	25
3.1.8.12. Caso de Uso: Jugar	25
3.1.8.12.1. Caso de Uso: Mover Fichas	27
3.1.8.12.2. Caso de Uso: Enviar	28
3.1.8.12.3. Caso de Uso: Cambiar	28
3.1.8.12.4. Caso de Uso: Pasar	28
3.1.8.12.5. Caso de Uso: Pausar	28
3.1.8.12.6. Caso de Uso: Abandonar Partida	29
3.1.8.12.7. Caso de Uso: Chatear	29

3.1.8.12.8. Caso de Uso: Cálculo de ELO	29
CAPÍTULO 4: DISEÑO	30
4.1. DIAGRAMA DE CLASES	30
4.1.1. DIAGRAMA DE CLASES: SERVIDOR	30
4.1.1.1. Clase Servidor	32
4.1.1.2. Clase ClienteHandler	32
4.1.1.3. Clase Enviador	32
4.1.1.4. Clase Recibidor	32
4.1.1.5. Clase Diccionario	33
4.1.1.6. Clase JugadorInfo	33
4.1.1.7. Clase Jugador	33
4.1.1.8. Clase Sala	33
4.1.1.9. Clase MesaInfo	33
4.1.1.10. Clase Mesa	34
4.1.1.11. Clase Partida	34
4.1.1.12. Clase Saco	34
4.1.1.13. Clase Tablero	34
4.1.1.14. Clase Celda	35
4.1.1.15. Clase Ficha	35
4.1.1.16. Clase Colocación	35
4.1.1.17. Clase Palabras	35
4.1.2. DIAGRAMA DE CLASES: CLIENTE	36
4.1.2.1. Clase Controlador	37
4.1.2.2. Clase Cliente	37
4.1.2.3. Clase Enviador	37
4.1.2.4. Clase Recibidor	37
4.1.2.5. Clase PanelJuego	37
4.1.2.6. Clase JugadorInfo	37
4.1.2.7. Clase Sala	38
4.1.2.8. Clase MesaInfo	38
4.1.2.9. Clase Partida	38
4.1.2.10. Clase Mesa	38
4.1.2.11. Clase Tablero	38
4.1.2.12. Clase Celda	39
4.1.2.13. Clase Ficha	39
4.1.2.14. Clase Colocación	39
4.1.2.15. Frames	39
4.1.3. PROTOCOLO	42
4.1.3.1. Clase Protocolo	42
4.1.3.2. El Interfaz Paquete	43
4.2. DIAGRAMAS DE SECUENCIA	78
4.2.1. Diagrama de Secuencia: Navegación	78
4.2.2. Diagrama de Secuencia: Entrar Mesa y Comienzo Partida	80
4.2.3. Diagrama de Secuencia: Jugar Partida	81
4.2.4. Diagrama de Secuencia: Cambiar Comodín	83
4.2.5. Diagrama de Secuencia: Acabar Partida	84
4.2.6. Diagrama de Secuencia: Pausar Partida	87
4.2.7. Diagrama de Secuencia: Abandonar	91
4.2.8. Diagrama de Secuencia: Salir del Sistema	91
CAPÍTULO 5: IMPLEMENTACIÓN Y PRUEBAS	94

5.1. IMPLEMENTACIÓN.....	94
5.2. DESPLIEGUE	100
5.3. PRUEBAS.....	103
CAPÍTULO 6: CONCLUSIONES Y LÍNEAS FUTURAS.....	104
6.1. CONCLUSIONES	104
6.2. LÍNEAS FUTURAS	105
CAPÍTULO 7: BIBLIOGRAFÍA.....	106
ANEXO: MANUAL DE USUARIO	108

Listado de Figuras

Ilustración 1.1 Arquitectura del Sistema	8
Ilustración 3.1 Diagrama de casos de uso general	18
Ilustración 3.2 Diagrama de caso de uso Jugar	21
Ilustración 3.3 Caso de uso Jugar	27
Ilustración 4.1 Diagrama de Clases Servidor	31
Ilustración 4.2 Diagrama de Clases Cliente	36
Ilustración 4.3 Diagrama de Clases Cliente Frames	40
Ilustración 4.4 Clase Protocolo	42
Ilustración 4.5 Interfaz Paquete	43
Ilustración 4.6 PaqueteError	44
Ilustración 4.7 PaqueteLogin	45
Ilustración 4.8 PaqueteSalon	46
Ilustración 4.9 PaqueteSelecSala	47
Ilustración 4.10 PaqueteSala	48
Ilustración 4.11 PaqueteChatSalon	49
Ilustración 4.12 PaqueteChatSala	50
Ilustración 4.13 PaqueteJugadorEntraSalon	51
Ilustración 4.14 PaqueteJugadorEntraSala	52
Ilustración 4.15 PaqueteVolverSalon	53
Ilustración 4.16 PaqueteSaleSala	54
Ilustración 4.17 PaqueteJugadorSeVa	55
Ilustración 4.18 PaqueteSelecMesa	56
Ilustración 4.19 PaqueteJugadorEntraMesa	57
Ilustración 4.20 PaqueteMesa	58
Ilustración 4.21 PaqueteConfig	59
Ilustración 4.22 PaqueteSalirMesa	60
Ilustración 4.23 PaqueteSaleMesa	61
Ilustración 4.24 PaqueteChatMesa	62
Ilustración 4.25 PaqueteActualizacion	63
Ilustración 4.26 PaqueteAceptarConfig	64
Ilustración 4.27 PaqueteConfigAceptada	65
Ilustración 4.28 PaqueteComenzarPartida	66
Ilustración 4.29 PaquetePausarJuego	67
Ilustración 4.30 PaqueteReanudarJuego	68
Ilustración 4.31 PaqueteJugadorMueveFicha	69
Ilustración 4.32 PaqueteTurno	70
Ilustración 4.33 PaqueteJugadorEnvia	71
Ilustración 4.34 PaqueteJugarFichas	72

Ilustración 4.35 PaqueteInfoJugada.....	73
Ilustración 4.36 PaqueteCambiar.....	74
Ilustración 4.37 PaquetePasar.....	75
Ilustración 4.38 PaqueteCambiaComodin.....	76
Ilustración 4.39 PaqueteAcabarPartida.....	77
Ilustración 4.40 Diagrama de Secuencia Navegación.....	79
Ilustración 4.41 Diagrama de Secuencia Entrar Mesa y Comienzo Partida.....	80
Ilustración 4.42 Diagrama de Secuencia Jugar Partida.....	82
Ilustración 4.43 Diagrama de Secuencia Cambiar Comodín.....	83
Ilustración 4.44 Diagrama de Secuencia Acabar Partida (Jugadores aceptan).....	85
Ilustración 4.45 Diagrama de Secuencia Acabar Partida (Jugador rechaza).....	86
Ilustración 4.46 Diagrama de Secuencia Pausar Partida: Jefe de Mesa pausa I.....	87
Ilustración 4.47 Diagrama de Secuencia Pausar Partida: Jefe de Mesa pausa II.....	88
Ilustración 4.48 Diagrama de Secuencia Pausar Partida: Jugador pausa I.....	89
Ilustración 4.49 Diagrama de Secuencia Pausar Partida: Jugador pausa II.....	90
Ilustración 4.50 Diagrama de Secuencia Abandonar.....	91
Ilustración 4.51 Diagrama de Secuencia Salir Sistema I.....	92
Ilustración 4.52 Diagrama de Secuencia Salir Sistema II.....	93
Ilustración 5.1 Declaración de las listas de clientes del sistema y jugadores del Salón.....	94
Ilustración 5.2 Declaración de la lista de jugadores de la Sala.....	95
Ilustración 5.3 Declaración de la lista de jugadores de la mesa.....	95
Ilustración 5.4 Implementación Clase Partida.....	95
Ilustración 5.5 Clase interna RemindTask.....	96
Ilustración 5.6 Métodos para iniciar, reiniciar y parar tiempo partida.....	97
Ilustración 5.7 Métodos para activar y desactivar la pausa.....	97
Ilustración 5.8 Clase Interna RemindTask y métodos para iniciar, parar y reiniciar el tiempo de pausa.....	98
Ilustración 5.9 Apertura de un socket con el host con la URL del directorio que contiene el applet, y el puerto.....	99
Ilustración 5.10 XAMPP: Ejecución de Apache y MySQL como servicio.....	100
Ilustración 5.11 XAMPP: Añadir contraseña para restringir acceso.....	101
Ilustración 5.12 Base de datos: Tabla Jugador.....	101
Ilustración 5.13 Base de datos: creación de un usuario.....	102
Ilustración 5.14 Ejecución del servidor de juego.....	102
Ilustración 5.15 Programa que comunica de los cambios de la IP.....	103
Ilustración A.1 Página Principal QUIJIBO.....	108
Ilustración A.2 Botón JUGAR.....	109
Ilustración A.3 Formulario de Registro.....	109
Ilustración A.4 Rankings de jugadores.....	110
Ilustración A.5 Acceso a la Cuenta de Jugador.....	111
Ilustración A.6 Cuenta de Jugador.....	112
Ilustración A.7 Búsqueda de Jugador por nick.....	113
Ilustración A.8 Logout.....	113
Ilustración A.9 Pantalla de Login.....	120
Ilustración A.10 El Salón Principal.....	121
Ilustración A.11 La Sala.....	122

Ilustración A.12 La Mesa	123
Ilustración A.13 Predicción de ganancia	128
Ilustración A.14 Diferencia de escalafón	129
Ilustración A.15 Final de Partida	130

CAPÍTULO 1: INTRODUCCIÓN

En este primer capítulo se realizará una pequeña introducción en la que se explicarán conceptos como los objetivos generales, se definirán los requisitos y una forma básica de la solución a emplear.

1.1. INTRODUCCIÓN

En los últimos años, los juegos online multijugador han experimentado una gran popularidad. Juegos de deportes, cartas, billar, juegos de mesa tradicionales... prácticamente todos tienen su versión.

Lo que los hace tan atractivos es la posibilidad de poder competir con cualquier otra persona, de una manera rápida y sencilla.

En este proyecto se ha desarrollado un juego/aplicación, basado en el “Scrabble”, uno de los más populares juegos de mesa de la historia, que actualmente cuenta con 3 campeonatos mundiales y millones de seguidores.

La figura 1.1 muestra la arquitectura del Sistema.

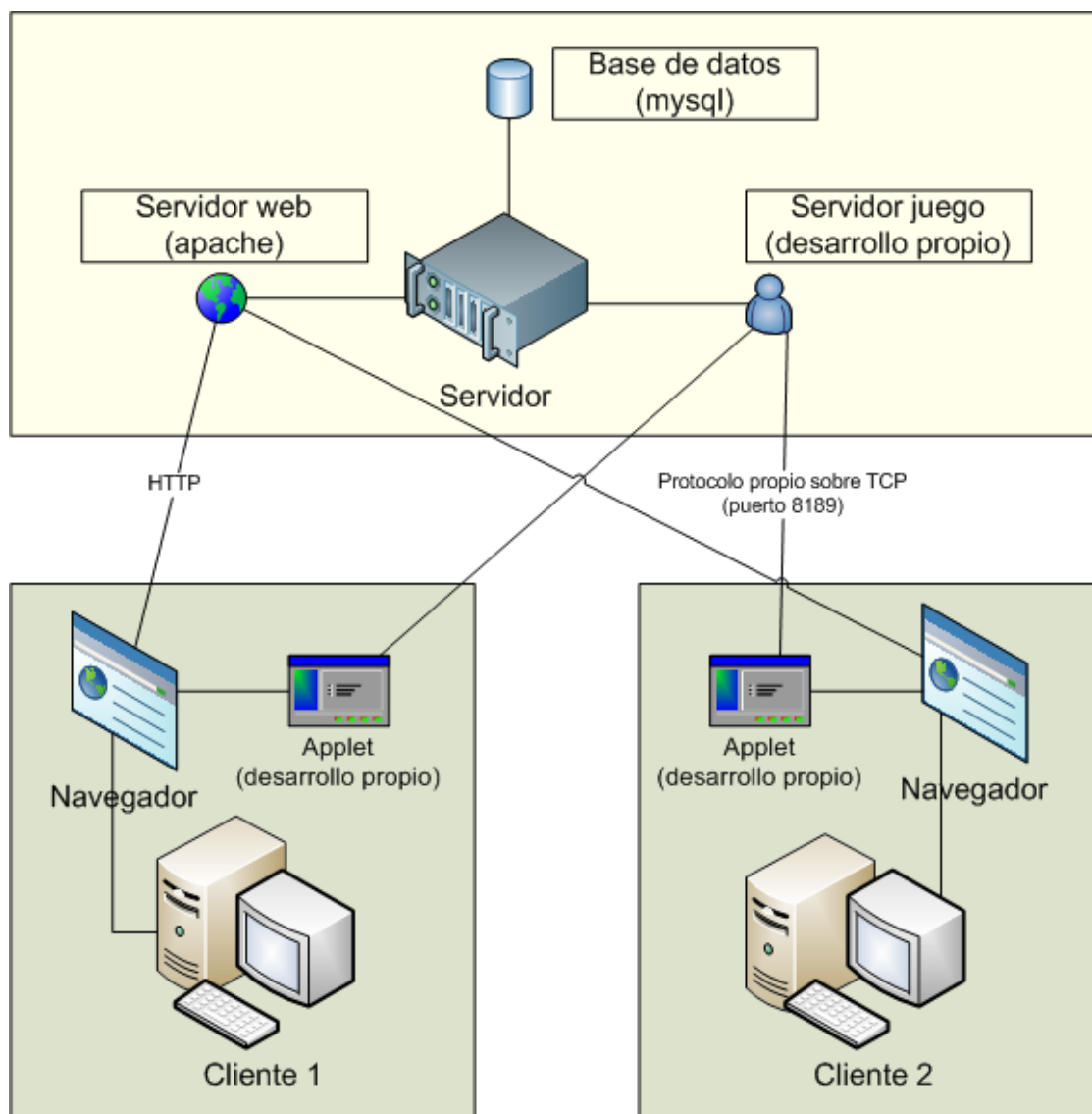


Ilustración 1.1 Arquitectura del Sistema

1.2. OBJETIVOS

El objeto de este proyecto es la realización de un sistema completo que permite a los usuarios poder jugar entre ellos a un juego basado en el popular juego de mesa “Scrabble” desde cualquier lugar. Por esto, se puso especial énfasis en la usabilidad, accesibilidad y compatibilidad de la aplicación que usarán los clientes.

Uno de los principales objetivos se trata de la comodidad para el usuario, dado que la aplicación no requiere instalación gracias a la utilización de Applets de Java. Asimismo, al estar programada en Java, se trata de una aplicación compatible con diferentes plataformas.

Los usuarios podrán jugar entre ellos sin necesidad de registrarse sólo por diversión o, si lo prefieren, podrán registrarse y disputar partidas para entrar en los rankings de los mejores jugadores, guardándose sus datos y estadísticas en una Base de Datos.

En resumen, todas estas características la convierten en una aplicación atractiva y sencilla de usar, para que los usuarios puedan pasar un rato agradable y divertido.

Por temas de Copyright, no se utilizará el nombre de “Scrabble”; se ha decidido llamar a la aplicación “QUIJIBO”, y a partir de ahora se hará referencia al juego por este nombre.

1.3. DESCRIPCIÓN DEL PROBLEMA

El acceso al juego se realizará a través de una página Web en cuyo contexto se ejecutará la aplicación y en la que el usuario podrá realizar las siguientes acciones:

- Registro de usuarios: El usuario podrá elegir un nombre de usuario o nick único para poder disputar partidas en modo competitivo.
- Acceder a una cuenta: Sólo para usuarios registrados. Será necesario introducir el nombre de usuario y la contraseña para poder acceder a esta sección. Una vez logueado, el usuario podrá ver sus estadísticas, así como buscar información de otros usuarios registrados.
- Consultar rankings de los mejores jugadores: Esta sección es visible para todo el mundo.
- Ayuda: Manual con la descripción del juego y las instrucciones para jugar.
- Jugar: Se ejecutará el Applet y el usuario podrá empezar a jugar como invitado o como usuario registrado introduciendo su nick y su contraseña.

En los siguientes apartados y a lo largo de la memoria se profundizará en el análisis, diseño e implementación de la aplicación, explicando detalladamente el proceso del problema.

1.4. REQUISITOS

En este apartado se mostrarán los requisitos mediante los cuales se ha hecho el análisis y diseño de la aplicación software. Se explicarán los requisitos generales y, a continuación, los requisitos por categorías.

1.4.1. Requisitos generales

Los usuarios podrán registrarse en la Web o no; en el caso de hacerlo, deberán introducir un nombre de usuario que no esté ya elegido y una contraseña.

1.4.2. Requisitos del Interfaz

La interfaz de usuario que se va a diseñar tanto para la Web como para el Applet, deberá ser intuitiva en todas sus secciones. La aplicación está pensada para que pueda ser utilizada por una amplia variedad de usuarios, desde aquellos que tienen un nivel alto, hasta los usuarios inexpertos en informática. Por eso mismo, las pautas para utilizar la aplicación serán fáciles de percibir y, en caso de dudas, existirá una sección de Ayuda, en la que se mostrará un manual de usuario. Los usuarios interactuarán de las siguientes maneras:

- Para un usuario que entra por primera vez en la página, podrá registrarse si así lo desea, aunque no es un requisito necesario para poder jugar.
- Para todos los usuarios que disponen ya de una cuenta, necesitarán loguearse si desean ver sus estadísticas y buscar datos de otros usuarios registrados.
- Cualquier usuario, disponga de una cuenta o no, podrá ejecutar la aplicación para comenzar a jugar. Una vez cargado el juego, se podrán realizar las siguientes acciones:
 - Los usuarios que deseen jugar como invitados, introducirán un nick que necesariamente deberá ser un nick no registrado y que no esté ya siendo usado por otro jugador.
 - Los usuarios que dispongan de una cuenta y deseen jugar para cambiar sus estadísticas, introducirán su nick y su contraseña. No importa que ya los hayan introducido en la Web, ya que ésta es independiente de la aplicación que se ejecuta en el Applet.

- Una vez introducido el nick (ya sean registrados o invitados), los usuarios entrarán automáticamente al Salón Principal, donde podrán ver en tiempo real al resto de jugadores del Salón (su nick, si son registrados o no y, en caso de ser registrados, su ELO y puntuación generales). Podrán comunicarse entre ellos mediante un chat, así como dispondrán de una lista de Salas a las que podrán acceder si su capacidad (que viene indicada también en tiempo real) no está completa.
- Elegida ya una Sala, el usuario entrará en ella, donde se encontrará un panorama muy similar: los jugadores de esa Sala, con los que podrá comunicarse, y una lista de Mesas a las que podrá acceder si no su capacidad (que viene indicada, así como los nicks de los jugadores que la ocupan) no está completa. Todo esto se mostrará en tiempo real. El usuario podrá regresar al Salón Principal si lo desea.
- Habiendo entrado ya a una mesa, la partida no comenzará hasta que la mesa no se haya llenado y todos los ocupantes hayan aceptado la configuración del juego.
- La mesa dispondrá de un tablero, un atril para las fichas, los botones necesarios para el juego, un panel de información de jugadas, de tiempo y de puntuación del jugador, una lista de jugadores de la mesa y su puntuación en la partida, y un chat. En el chat, los usuarios podrán comunicarse con los usuarios de la Sala hasta que comience la partida, momento en el que sólo se podrán comunicar exclusivamente con los usuarios de la mesa.
- Para que la partida se dispute en Modo Competición y los resultados se vean reflejados en las estadísticas de los jugadores, es necesario que absolutamente todos los jugadores de la mesa sean usuarios registrados. En caso contrario, la partida se disputará en modo no competitivo, y no puntuará en la Clasificación General.
- Terminada la partida, los datos se guardarán en caso de haber sido disputada en Modo Competición. En cualquier caso, se dará la opción se empezar una nueva partida, que los usuarios podrán aceptar o no. Si un usuario no acepta, volverá a la Sala
- Un usuario siempre podrá abandonar la mesa y volver a la Sala, incluso durante el transcurso de una partida, aunque en este caso si es un usuario registrado disputando una partida en Modo Competición, será penalizado y así se verá reflejado en sus estadísticas.
- Un usuario siempre podrá salir de la aplicación cuando lo desee.

· ELO: Se ha mencionado que los jugadores registrados podrán mantener y modificar un coeficiente ELO, que determina la “fuerza” de un jugador. En el Manual de Usuario se detalla la información acerca de su importancia y las fórmulas para calcularlo.

1.4.3. Requisitos de Hardware y de Sistema

Para realizar este proyecto será necesario un equipo con la plataforma de desarrollo NetBeans. Así mismo, se requerirá de un servidor de bases de datos (MySQL), aplicaciones de administración de la base de datos y un servidor Web (Apache). Sería interesante contar con un equipo accesible desde fuera de la red de la universidad (al menos al puerto Web y al del servidor de juegos).

1.4.4. Requisitos de la Base de Datos

De cada usuario registrado se guardarán los siguientes datos:

- Nombre de usuario.
- Contraseña.
- Fecha de registro.
- Puntuación general.
- ELO.
- Partidas jugadas.
- Partidas ganadas.
- Partidas abandonadas.

En un principio no será necesario que la base de datos sea muy grande.

1.5. SOLUCIÓN PROPUESTA

La solución que se ha propuesto para este proyecto ha sido la realización de una página Web donde se podrá ejecutar la aplicación mediante el uso de Applets. Así pues, el proyecto se divide en dos partes:

- La realización de la aplicación del juego se trata de la parte más importante y fundamental. Será necesario implementar una arquitectura cliente-servidor en la que se desarrollan tanto el cliente como el servidor en su totalidad. Se desarrollará también sobre TCP un protocolo de comunicación propio.
- El desarrollo de una página Web donde los usuarios puedan registrarse y consultar datos, así como acceder a la mencionada aplicación.

Además de esto, en la Web, los usuarios registrados podrán consultar sus datos y los de otros usuarios registrados; también se mantendrá una clasificación general de los mejores jugadores. La página leerá datos de la base de datos en la que estarán almacenados los datos y estadísticas sobre los usuarios.

1.5.1. Lenguaje de Programación

Para el desarrollo de este proyecto se usará principalmente Java, además de HTML, PHP, CSS y JavaScript. Como ya se ha comentado, la realización del proyecto se divide en dos partes bien diferenciadas.

- Para la realización del juego (la parte del cliente y del servidor), se decidió utilizar Java sin ninguna duda, por diversas razones:
 - La programación orientada a objetos.
 - La comodidad que supone trabajar con sockets y streams en Java para enviar y recibir datos en red.
 - La necesidad de utilizar hilos para las conexiones cliente-servidor.
 - La utilización de Applets, que permiten la ejecución de la aplicación sin necesidad de que el usuario tenga que descargar nada.
 - La facilidad para crear interfaces gráficas de usuario.
 - Poder trabajar con bases de datos.
 - La compatibilidad con diferentes plataformas.
- En cuanto a la parte Web, se utiliza una combinación de HTML, PHP, y JavaScript:
 - HTML, para el diseño de la página.
 - CSS, para definir los estilos de la página.
 - PHP: Principalmente para poder trabajar con la base de datos.
 - JavaScript: Se emplea una pequeña parte de JavaScript para la ejecución del juego en un Applet embebido en una nueva ventana.

1.5.2. Base de Datos

La base de datos escogida ha sido MySQL. Se trata de un sistema gestor de base de datos relacional, multihilo y multiusuario, desarrollado como software libre en un esquema de licenciamiento dual.

Existen varias API que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo Java (con una implementación nativa del driver de Java).

Su popularidad como aplicación Web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

CAPÍTULO 2: HERRAMIENTAS

2.1. NetBeans IDE

NetBeans IDE es un entorno de desarrollo visual de código abierto para aplicaciones programadas mediante Java, uno de los lenguajes de programación más poderosos del momento.

Su aprendizaje se ha convertido en fundamental para quienes están interesados en el desarrollo de aplicaciones multiplataforma.

No importa que la máquina donde se instale sea un PC con Linux o Windows o un Mac o un Windows, pues el funcionamiento del programa creado será igual gracias al intérprete de Java (el cual sí será distinto para cada plataforma).

Con NetBeans IDE no solo es posible elaborar potentes aplicaciones para el Escritorio, también para la Web y para dispositivos portátiles, como móviles o Pocket PC, sin que cambie la forma de programar.

La programación mediante NetBeans se realiza a través de componentes de software modulares, también llamados módulos.

2.2. XAMPP

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl.

El nombre proviene del acrónimo de **X** (para cualquiera de los diferentes sistemas operativos), **A**pache, **M**ySQL, **P**HP, **P**erl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas.

Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X.

XAMPP solamente requiere descargar y ejecutar un archivo, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin.

Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado.

2.3. Photoshop

Adobe Photoshop es una aplicación informática en forma de taller de pintura y fotografía que trabaja sobre un "*lienzo*" y que está destinado para la edición, retoque fotográfico y pintura a base de imágenes de mapa de bits (o gráficos rasterizados).

Es un producto elaborado por la compañía de software Adobe Systems, inicialmente para computadores Apple pero posteriormente también para plataformas PC con sistema operativo Windows.

Photoshop en sus versiones iniciales trabajaba en un espacio bitmap formado por una sola capa, donde se podían aplicar toda una serie de efectos, textos, marcas y tratamientos. En la actualidad lo hace con múltiples capas.

A medida que ha ido evolucionando el software ha incluido diversas mejoras fundamentales, como la incorporación de un espacio de trabajo multicapa, inclusión de elementos vectoriales, gestión avanzada de color (ICM/ICC), tratamiento extensivo de tipografías, control y retoque de color, efectos creativos, posibilidad de incorporar *plugins* de terceras compañías, exportación para Web entre otros.

Photoshop se ha convertido, casi desde sus comienzos, en el estándar *de facto* en retoque fotográfico, pero también se usa extensivamente en multitud de disciplinas del campo del diseño y fotografía, como diseño Web, composición de imágenes bitmap, estilismo digital, fotocomposición, edición y grafismos de vídeo y básicamente en cualquier actividad que requiera el tratamiento de imágenes digitales.

Aunque el propósito principal de Photoshop es la edición fotográfica, este también puede ser usado para crear imágenes, efectos, gráficos y más con muy buena calidad.

CAPÍTULO 3: ANÁLISIS DEL PROBLEMA

Una vez expuesta la introducción al problema, donde se ha indicado qué es lo que se desea implementar, hay que centrarse en el análisis.

Para ello se realizará un estudio sobre los casos de uso que aparezcan a lo largo del problema. Los casos de uso son descripciones de la funcionalidad asociada al sistema, pero con la ventaja de que son totalmente independientes a las siguientes fases del desarrollo (concretamente, independiente de la fase de implementación). Es decir, utilizar los diagramas de casos de uso en el análisis del problema no va unido a la implementación del proceso, precisamente el análisis de casos de uso es una representación del lenguaje natural entendible por todo tipo de usuario, donde se hace una descripción detallada de todas las funcionalidades del Sistema que se quiere diseñar.

3.1. DIAGRAMA DE CASOS DE USO: General

Tras haber recogido todos los requisitos en los apartados anteriores (Introducción y Análisis), y tras haber realizado una entrevista con el hipotético usuario de la aplicación, se comenzará a mostrar los diagramas de casos de uso que se han obtenido.

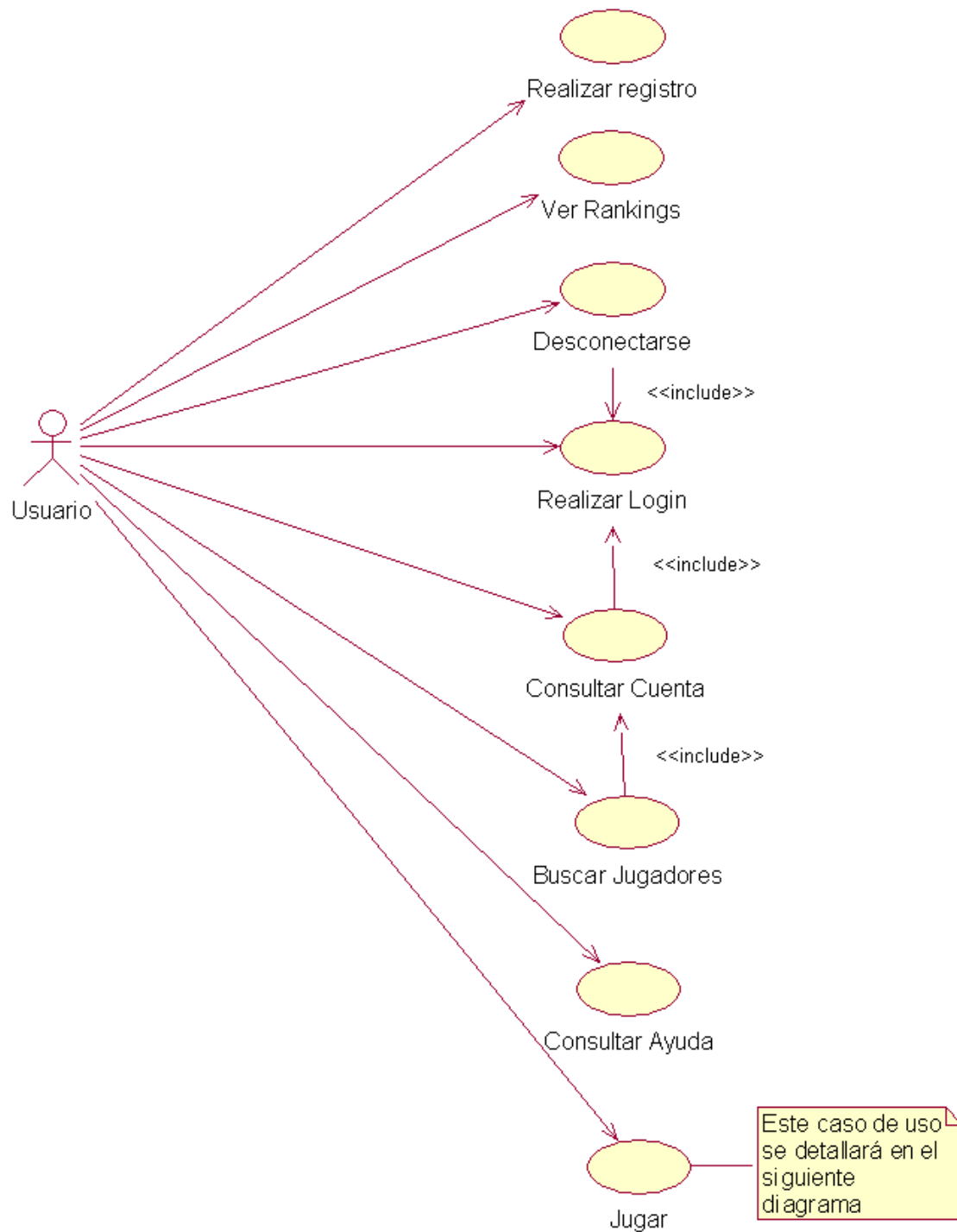


Ilustración 3.1 Diagrama de casos de uso general

3.1.1. Caso de Uso: Realizar Registro

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá registrarse si desea jugar como Usuario Registrado, y así guardar un acumulado de puntos y su ELO. Necesitará introducir un nick único y una contraseña.

3.1.2. Caso de Uso: Ver Rankings

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá ver una clasificación general de QUIJIBO, en dos listas que mostrarán los diez mejores jugadores, atendiendo aquellos que tengan una mayor puntuación y el ELO más alto, respectivamente.

3.1.3. Caso de Uso: Realizar Login

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá realizar el login si está registrado para acceder a determinadas secciones de la página.

3.1.4. Caso de Uso: Desconectarse

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá desconectarse de su cuenta en todo momento.

3.1.5. Caso de Uso: Consultar Cuenta

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá consultar sus datos y estadísticas, habiendo realizado previamente el Login.

3.1.6. Caso de Uso: Buscar Jugadores

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá buscar a cualquier jugador registrado introduciendo su nick y ver sus datos y estadísticas de juego.

3.1.7. Caso de Uso: Consultar Ayuda

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario tendrá a su disposición un manual de ayuda para utilizar la aplicación y consultar las reglas del juego.

3.1.8. Caso de Uso: Jugar

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá acceder al juego desde la página Web y comenzar a jugar. Este caso de uso se detallará a continuación.

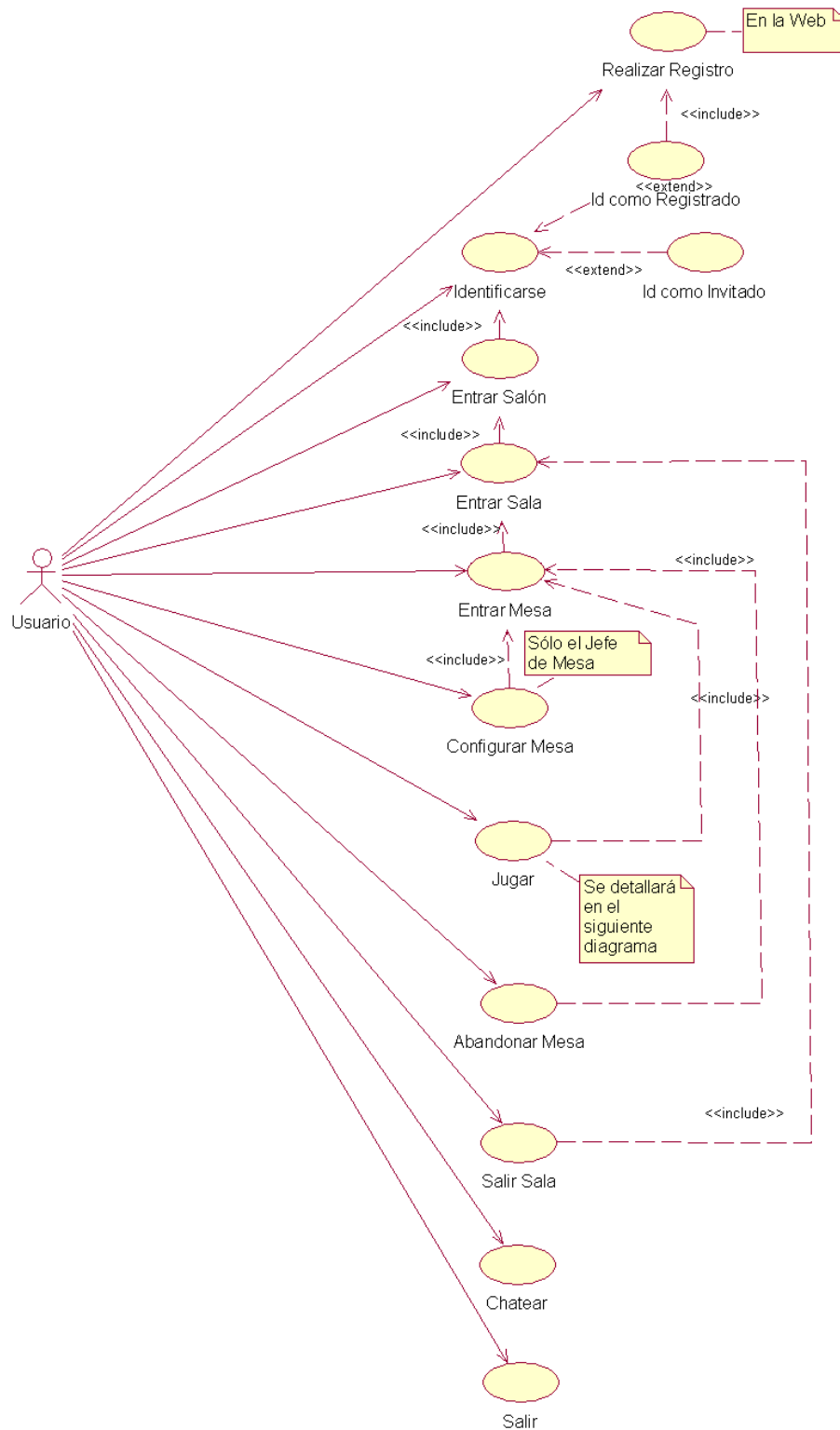


Ilustración 3.2 Diagrama de caso de uso Jugar

3.1.8.1. Caso de Uso: Identificarse

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario tendrá que utilizar un nick para poder jugar, ya sea como invitado o como usuario registrado.

3.1.8.2. Caso de Uso: Id como Registrado

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario, si dispone de una cuenta, deberá introducir su nick y su contraseña.

3.1.8.3. Caso de Uso: Id como Invitado

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario introducirá un nick, que debe cumplir:

- No ser un nick que pertenezca a un usuario registrado.
- En el caso de ser un nick no registrado, no debe estar siendo utilizado ya por otro usuario que se encuentre en el juego.

3.1.8.4. Caso de Uso: Entrar Salón

• **Actores:**

El usuario de la aplicación.

•Descripción:

Una vez logueado, el usuario será dirigido automáticamente al Salón Principal, donde se mostrará en tiempo real:

- La lista de Salas, con su capacidad máxima y el número de jugadores que hay en ella. El usuario tendrá que elegir una de estas salas.
- La lista de jugadores del Salón Principal, mostrándose si son invitados o registrados y, en este caso, su ELO y su puntuación general.
- Un cuadro de chat, donde podrá comunicarse con el resto de jugadores del Salón Principal.

3.1.8.5. Caso de Uso: Entrar Sala

• Actores:

El usuario de la aplicación.

•Descripción:

Una vez elegida la sala, si no está llena, el usuario entrará en ella, donde se mostrará en tiempo real:

- La lista de Mesas, con su capacidad máxima y el número de jugadores que hay en ella, así como sus nicks. El usuario tendrá que elegir una mesa para poder jugar.
- La lista de jugadores de la Sala, mostrándose si son invitados o registrados, y, en este caso, su ELO y su puntuación general.
- Un cuadro de chat, donde podrá comunicarse con el resto de jugadores del Sala.

3.1.8.6. Caso de Uso: Entrar Mesa

• Actores:

El usuario de la aplicación.

•Descripción:

Una vez elegida la mesa, si no está llena, el usuario entrará en ella, donde se mostrará:

- La lista de jugadores de la Mesa y su puntuación en la partida.
- El tablero.
- Un atril para las fichas.
- Los botones necesarios para el juego: Enviar, Cambiar, Pasar.

- Los botones Pausar y Abandonar.
- Un panel de información de jugadas.
- Un panel de tiempo.
- Un panel del puntuación.
- Un cuadro de chat. En el chat, los usuarios podrán comunicarse con los usuarios de la Sala hasta que comience la partida, momento en el que sólo se podrán comunicar exclusivamente con los usuarios de la mesa.

3.1.8.7. Caso de Uso: Configurar Mesa

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El modo de juego sólo puede ser configurado por el Jefe de Mesa antes de que empiece la partida. Se designa automáticamente como Jefe de Mesa al usuario que lleva más tiempo en la mesa (ya sea porque ha entrado el primero a una mesa vacía, o porque todos los jugadores que estaban antes que él hayan abandonado la mesa).

3.1.8.8. Caso de Uso: Abandonar Mesa

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá abandonar la mesa haya empezado una partida o no. En el caso de que la partida haya comenzado, y si ésta se estaba disputando en Modo Competición, el usuario será penalizado.

3.1.8.9. Caso de Uso: Salir Sala

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá regresar en todo momento al Salón Principal.

3.1.8.10. Caso de Uso: Chatear

• Actores:

El usuario de la aplicación.

• Descripción:

El usuario podrá comunicarse con el resto de usuarios a través de los chats del Salón, las Salas y las Mesas.

- Chat del Salón: El usuario puede comunicarse con los usuarios que se encuentren en el Salón Principal.
- Chat de la Sala: El usuario puede comunicarse con los usuarios que se encuentren en la misma Sala que él.
- Chat de la Mesa: El usuario que se encuentre en una Mesa puede comunicarse con los usuarios que se encuentren en la misma Sala que él mientras no haya comenzado la partida. Una vez comience, sólo podrá comunicarse con los jugadores que están en su misma mesa.

3.1.8.11. Caso de Uso: Salir

• Actores:

El usuario de la aplicación.

• Descripción:

El usuario podrá jugar salir del sistema en cualquier momento.

3.1.8.12. Caso de Uso: Jugar

• Actores:

El usuario de la aplicación.

• Descripción:

El usuario podrá jugar una partida una vez que se haya completado el número de jugadores necesarios y todos hayan aceptado la configuración de la partida. La partida podrá disputarse en:

- Modo Competición: La partida se disputará automáticamente en este modo si todos los jugadores de la mesa son usuarios registrados. De esta forma, al final de la partida, para cada jugador se guardarán las modificaciones en el ELO y la puntuación global.
- Modo no competitivo: Si al menos uno de los jugadores de la mesa es un Invitado, la partida no se disputará en Modo Competición, y los jugadores no verán modificados ni su ELO ni su puntuación global al final de la partida.

Este caso de uso se detallará a continuación:

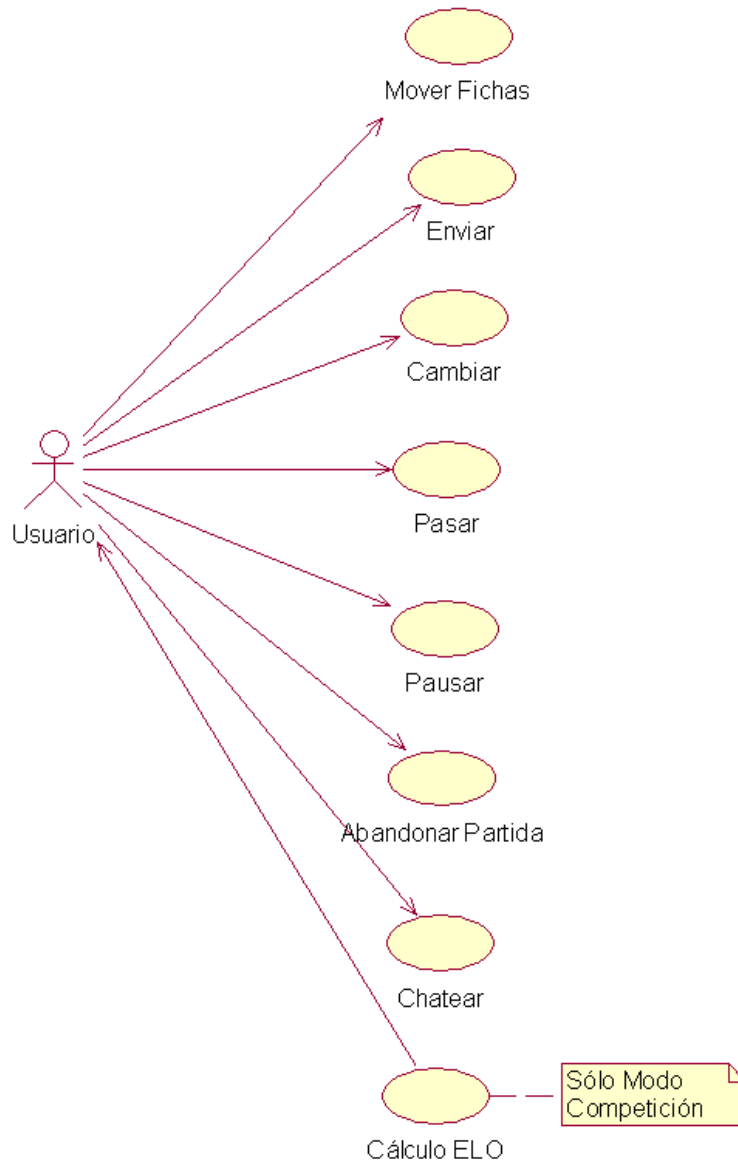


Ilustración 3.3 Caso de uso Jugar

3.1.8.12.1. Caso de Uso: Mover Fichas

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá mover las fichas de su atril al tablero y de vuelta a su atril cuando sea su turno.

3.1.8.12.2. Caso de Uso: Enviar

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá indicar mediante esta acción que desea jugar las fichas que ha colocado sobre el tablero.

3.1.8.12.3. Caso de Uso: Cambiar

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá indicar mediante esta acción que desea intercambiar las fichas que haya seleccionado en su atril cuando sea su turno.

3.1.8.12.4. Caso de Uso: Pasar

• **Actores:**

El usuario de la aplicación.

• **Descripción:**

El usuario podrá indicar mediante esta acción que desea jugar las fichas que ha colocado sobre el tablero cuando sea su turno.

3.1.8.12.5. Caso de Uso: Pausar

• **Actores:**

El usuario de la aplicación.

• Descripción:

El usuario podrá solicitar hasta tres pausas de un minuto a lo largo de la partida. Si el usuario que solicita la pausa es Jefe de Mesa, la pausa tendrá lugar inmediatamente. Si no lo es, el jefe de Mesa deberá aceptar la solicitud de Pausa.

3.1.8.12.6. Caso de Uso: Abandonar Partida**• Actores:**

El usuario de la aplicación.

• Descripción:

El usuario podrá abandonar la mesa durante el transcurso de una partida. Si la partida se estaba disputando en Modo Competición, el usuario será penalizado y esta información aparecerá en sus estadísticas de juego.

3.1.8.12.7. Caso de Uso: Chatear**• Actores:**

El usuario de la aplicación.

• Descripción:

Una vez comenzada la partida, el usuario sólo podrá comunicarse con los jugadores que están en su misma mesa.

3.1.8.12.8. Caso de Uso: Cálculo de ELO**• Actores:**

El usuario de la aplicación.

• Descripción:

Una vez finalizada la partida, habiéndose disputado en Modo Competición, se procederá al cálculo de ELO de cada uno de los jugadores, y se actualizará dicha información.

CAPÍTULO 4: DISEÑO

En este capítulo se presentarán las clases que definen el Sistema para ayudarnos en su descripción estática, se utilizarán diagramas de clases. Para reflejar el comportamiento dinámico, utilizaremos diagramas de secuencia para describir los escenarios más interesantes.

4.1. DIAGRAMA DE CLASES

4.1.1. DIAGRAMA DE CLASES: SERVIDOR

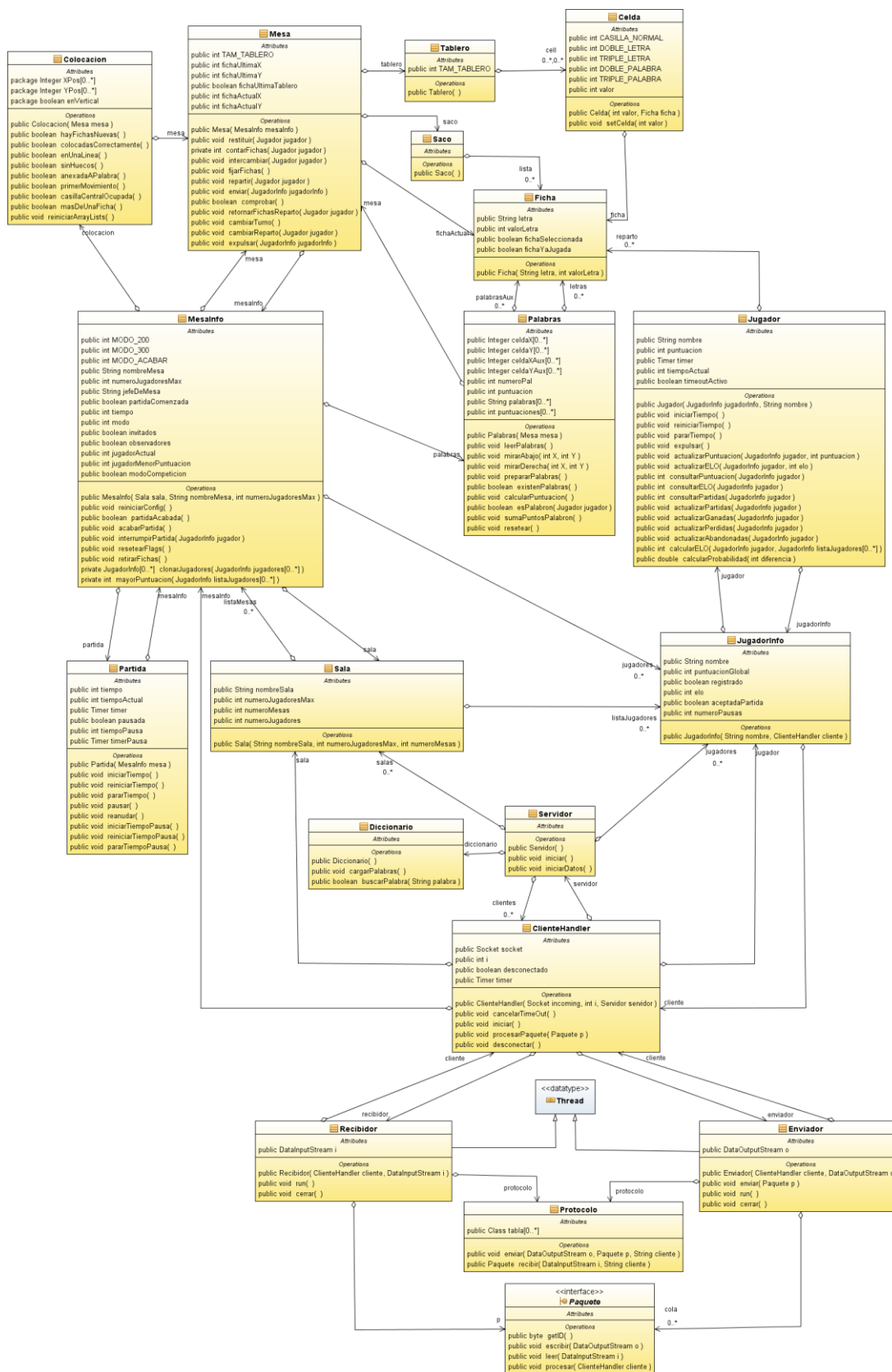


Ilustración 4.1 Diagrama de Clases Servidor

4.1.1.1. Clase Servidor

Esta clase se encarga de:

- Escuchar peticiones de conexión y de aceptarlas.
- Mantener la lista de clientes del Sistema
- Mantener la lista de jugadores del Salón Principal
- Mantener la lista de salas (inicializando el número de salas, su nombre, su capacidad máxima y el número de mesas por sala).
- Cargar el diccionario.

4.1.1.2. Clase ClienteHandler

Esta clase se encarga de:

- Crear un dataInputStream que usa un input stream para leer bytes del socket.
- Crear un dataOutputStream que usa un output stream para escribir bytes en el socket.
- Crear dos hilos por cada cliente (Enviador y Recibidor) que serán los encargados de enviar y recibir paquetes, e iniciar su ejecución.
- Realizar las acciones oportunas cuando se produzca la desconexión de un cliente.

4.1.1.3. Clase Enviador

Esta clase, que hereda de Thread, se encarga fundamentalmente de enviar paquetes de una cola bloqueante utilizando el Protocolo, que se verá más adelante.

4.1.1.4. Clase Recibidor

Esta clase, que hereda de Thread, se encarga de recibir paquetes por medio del Protocolo y de realizar las acciones asociadas a ellos.

4.1.1.5. Clase Diccionario

Esta clase se encargará de meter todas las palabras de un fichero en una hashtable, para después poder comprobar si una determinada palabra es una clave de la hashtable.

4.1.1.6. Clase JugadorInfo

Esta clase se encargará de mantener información general del cliente como jugador, como su nick, si es usuario registrado o no, su puntuación global, su ELO...

4.1.1.7. Clase Jugador

Esta clase guarda una estrecha relación con JugadorInfo; se encarga de mantener la información de un jugador en una partida, tal como la puntuación de la partida, las fichas que se le asignan...

Además consta de determinados métodos para, en el caso de tratarse de un jugador registrado, consultar y actualizar sus datos en la base de datos (puntuación global, ELO, partidas jugadas, ganadas, perdidas, abandonadas...).

4.1.1.8. Clase Sala

Esta clase se encarga de:

- Mantener una lista de jugadores de la sala.
- Mantener una lista de mesas de la sala (inicializando el número de mesas, su nombre, su capacidad máxima y el número de mesas por sala).

4.1.1.9. Clase MesaInfo

Esta clase se encarga fundamentalmente de mantener información sobre la mesa, tal como el nombre de la mesa, el número de jugadores requeridos, si ha comenzado una partida, la lista de jugadores de la mesa, el Jefe de Mesa, el modo de juego y tiempo por turno, si la partida se disputa en modo Competición, si se permiten invitados...

Como se ha mencionado se mantendrá una lista de jugadores de la mesa.

Además, se encarga de:

- Comprobar si una partida ha terminado, y realizar las acciones necesarias en este caso.
- Restablecer el tablero, el saco y los repartos, así como las puntuaciones y el tiempo de la partida cuando va a comenzar una partida.
- Realizar las acciones oportunas cuando se interrumpe una partida.

4.1.1.10. Clase Mesa

Esta clase guarda una estrecha relación con MesaInfo. Constará de determinados métodos que realizarán las acciones necesarias para el desarrollo del juego:

- Repartir fichas aleatoriamente.
- Comprobar una jugada.
- Fijar las fichas en el tablero y restituir el reparto de un jugador si la jugada es correcta.
- Devolver las fichas al reparto si una jugada es incorrecta o se pasa el turno del jugador.
- Intercambiar fichas.
- Cambiar de Turno.
- Las acciones necesarias si un jugador abandona la mesa.

4.1.1.11. Clase Partida

Esta clase se encarga del tiempo de la partida, los turnos y su duración, así como de las posibles pausas que puedan ser solicitadas.

4.1.1.12. Clase Saco

Esta clase consta de una lista de las fichas por repartir. Esta lista contiene todas las fichas del juego al inicio de la partida y conforme éstas se vayan repartiendo, se eliminarán de la lista. De esta forma se simulará la repartición de las fichas.

4.1.1.13. Clase Tablero

Esta clase consta de una matriz de celdas 15x15, asignándole a cada celda un determinado valor según su disposición en el tablero.

4.1.1.14. Clase Celda

Esta clase tiene el propósito de simular las casillas del tablero y del reparto: una celda podrá tener un determinador valor (Casilla Normal, Multiplicador de Letra por dos, Multiplicador de Letra por tres, Multiplicador de Palabra por dos, Multiplicador de Palabra por tres), y podrá estar ocupada o no por una determinada ficha.

4.1.1.15. Clase Ficha

Esta clase se encarga de simular las fichas del juego: una ficha tendrá un determinada letra asignada y un valor. Además, una ficha podrá o no haber sido jugada (ya se ha construido una palabra con la ficha sobre el tablero) o tratarse de una ficha que se encuentra seleccionada.

4.1.1.16. Clase Colocación

Esta clase se encarga de comprobar si las fichas sobre el tablero están correctamente colocadas (siguiendo las normas del juego: ocupando la casilla central si se trata de la primera jugada; anexada a una palabra ya existente, sin huecos y que las nuevas fichas estén en línea).

4.1.1.17. Clase Palabras

Esta clase se encarga de:

- Obtener la/s nueva/s palabra/s formada/s.
- Calcular la puntuación obtenida por cada una de ellas, atendiendo al valor de las fichas y de las casillas del tablero.
- Comprobar si una jugada ha sido “Palabrón”.
- Calcular el total de puntos obtenidos en la jugada.

4.1.2. DIAGRAMA DE CLASES: CLIENTE

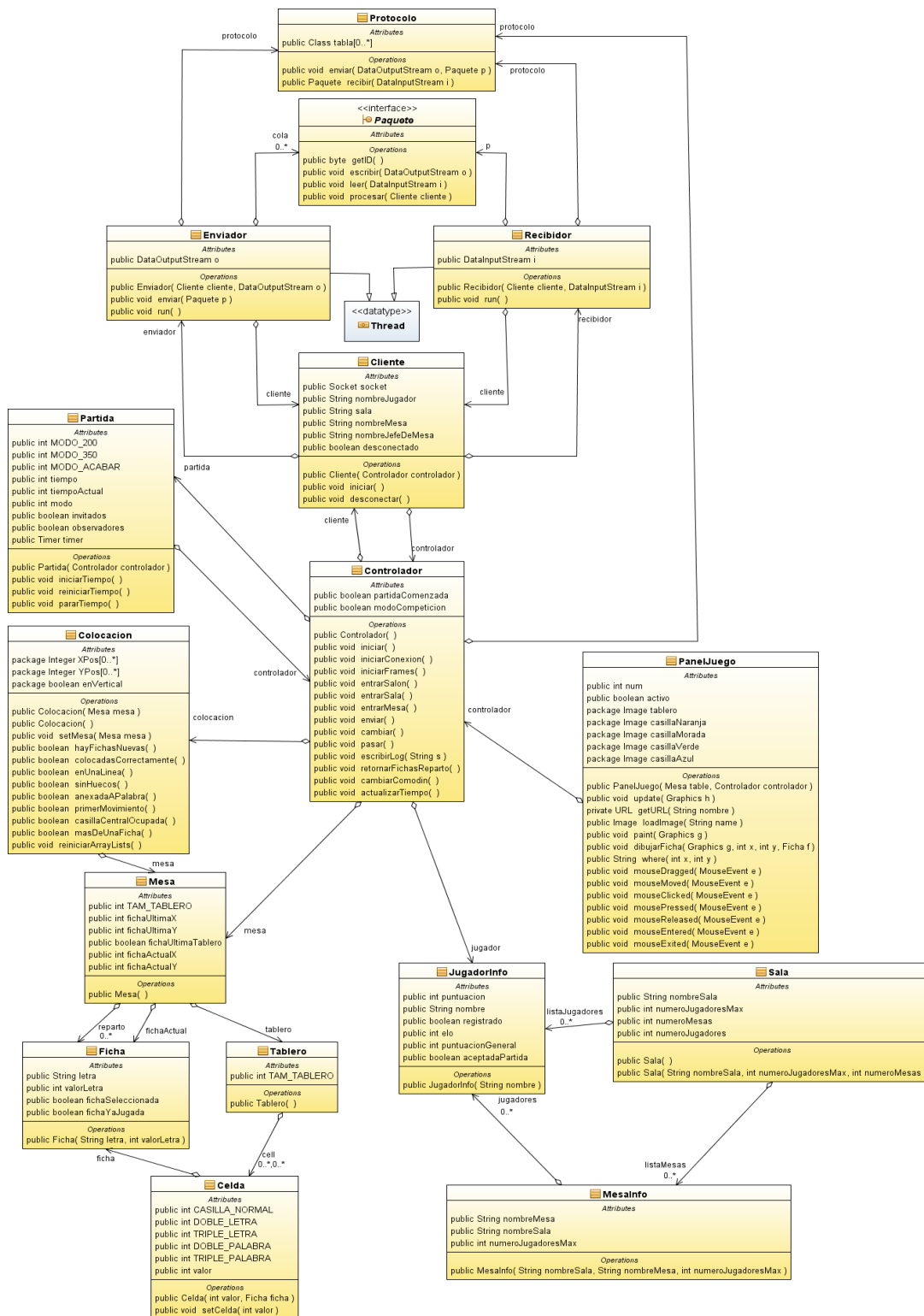


Ilustración 4.2 Diagrama de Clases Cliente

4.1.2.1. Clase Controlador

Se trata de una clase desde la que se tiene acceso a todas las demás (para inicializar todas las demás clases y para que éstas tengan acceso entre ellas).

4.1.2.2. Clase Cliente

Esta clase se utiliza para crear un socket y realizar una petición de conexión al Servidor, mediante el host y el puerto indicados.

También se encarga de cerrar el socket si se cierra la conexión.

4.1.2.3. Clase Enviador

Esta clase, que hereda de Thread, se encarga fundamentalmente de enviar paquetes de una cola bloqueante utilizando el Protocolo, que se verá más adelante.

4.1.2.4. Clase Recibidor

Esta clase, que hereda de Thread, se encarga de recibir paquetes por medio del Protocolo y de realizar las acciones asociadas a ellos.

4.1.2.5. Clase PanelJuego

Esta clase, que hereda de JPanel e implementa MouseMotionListener y MouseListener, se encarga de realizar el dibujo del tablero y del atril y de procesar correctamente los eventos del ratón para lograr emular la funcionalidad del juego ajustándose al estado de la partida.

4.1.2.6. Clase JugadorInfo

Esta clase contiene información que le llega del Servidor sobre el cliente como jugador: el nombre, si es registrado o no, la puntuación general y de la partida, ELO...

4.1.2.7. Clase Sala

Esta clase contiene información que le llega del Servidor sobre una determinada sala: el nombre de la sala, la capacidad máxima, la lista de jugadores de la sala y la lista de las mesas...

4.1.2.8. Clase MesaInfo

Esta clase contiene información que le llega del Servidor sobre una determinada mesa: el nombre de la mesa y de la sala en la que se encuentra, el número de jugadores necesarios y una lista de jugadores que se encuentren en la mesa.

4.1.2.9. Clase Partida

Esta clase se utiliza para que al jugador se le muestre un contador del tiempo de cada turno.

El contador se iniciará cuando le llegue el paquete que indica el comienzo de la partida del Servidor o el paquete que indica que se reanuda el juego tras una pausa.

El contador se reiniciará cada vez que le llegue un paquete de cambio de turno.

El contador parará cuando al cliente le llegue un paquete que indique el fin de la partida, el abandono de un jugador o una pausa en la partida.

4.1.2.10. Clase Mesa

Esta clase consta de un tablero y un reparto, que será lo que el usuario verá cuando esté en la mesa. Se irá actualizando con los paquetes que le lleguen del Servidor.

4.1.2.11. Clase Tablero

Esta clase consta de una matriz de celdas 15x15, asignándole a cada celda un determinado valor según su disposición en el tablero.

4.1.2.12. Clase Celda

Esta clase tiene el propósito de simular las casillas del tablero y del reparto: una celda podrá tener un determinador valor (Casilla Normal, Multiplicador de Letra por dos, Multiplicador de Letra por tres, Multiplicador de Palabra por dos, Multiplicador de Palabra por tres), y podrá estar ocupada o no por una determinada ficha.

4.1.2.13. Clase Ficha

Esta clase se encarga de simular las fichas del juego: una ficha tendrá un determinada letra asignada y un valor. Además, una ficha podrá o no haber sido jugada (ya se ha construido una palabra con la ficha sobre el tablero), o tratarse de una ficha que se encuentra seleccionada.

4.1.2.14. Clase Colocación

Esta clase se encarga de comprobar si las fichas sobre el tablero están correctamente colocadas (siguiendo las normas del juego: ocupando la casilla central si se trata de la primera jugada; anexada a una palabra ya existente, sin huecos y que las nuevas fichas estén en línea).

La jugada se enviará al Servidor sólo si las fichas están correctamente colocadas.

4.1.2.15. Frames

El siguiente diagrama muestra los frames utilizados y la navegabilidad:

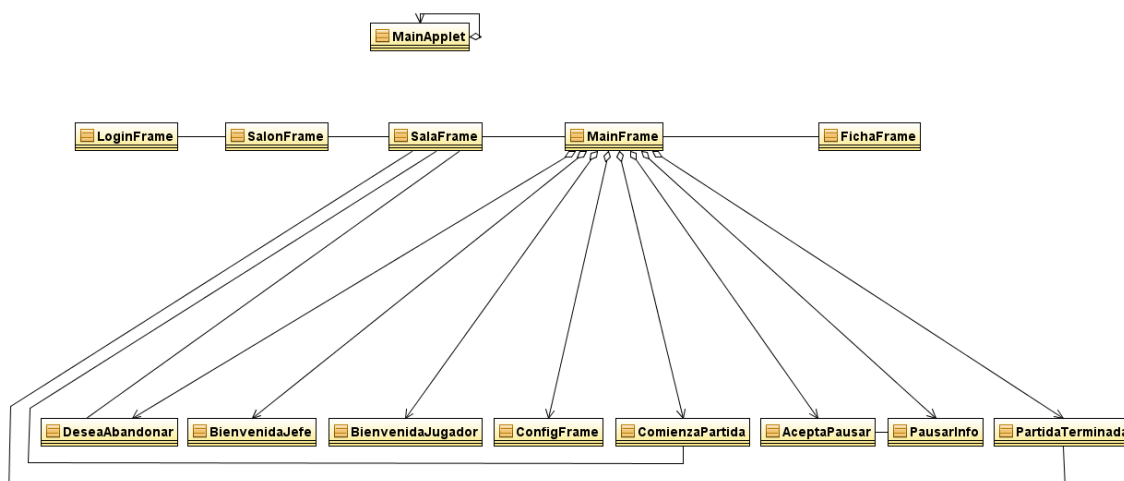


Ilustración 4.3 Diagrama de Clases Cliente Frames

- LoginFrame: El usuario podrá loguearse como invitado o usuario registrado utilizando contraseña. Una vez logueado correctamente, pasará a SalonFrame.
- SalonFrame: En este frame se mostrará una tabla con las salas (nombre, capacidad máxima y número actual de jugadores), una tabla de jugadores del Salón (su nick, si son registrados o no, ELO y puntuación global) y un chat del Salón (textArea, textField). Seleccionando una sala y pulsando “Entrar”, el jugador pasará a SalaFrame si su capacidad no está completa.
- SalaFrame: En este frame se mostrará una tabla con las mesas (nombre, número de jugadores necesarios, número actual de jugadores), de jugadores de la Sala (su nick, si son registrados o no, ELO y puntuación global) y un chat de la Sala (textArea, textField). Seleccionando una mesa, se mostrará en una tabla los nombres de los jugadores que la ocupan. Seleccionando una mesa y pulsando “Entrar”, el jugador pasará a MainFrame si su capacidad no está completa. El jugador pasará a SalonFrame si pulsa “Volver”.
- MainFrame: En este frame se mostrará en un panel un tablero y un atril para las fichas; los botones necesarios para el juego, un panel de información de jugadas, de tiempo y de puntuación del jugador, una tabla de jugadores de la mesa y su puntuación en la partida y un chat (textArea, textField). Pulsando “Abandonar”, regresará a SalaFrame.
- FichaFrame: Este frame se mostrará cuando se pulse con el botón derecho sobre una ficha comodín no jugada que esté sobre el tablero. En el frame se mostrarán tantos botones como letras distintas haya en el juego; pulsando el botón correspondiente la ficha comodín adquirirá el valor de esa letra y se ocultará el frame.

- BienvenidaJefe: Este frame se mostrará al Jefe de Mesa (informándole de ello) mientras la mesa no esté completa.
- BienvenidaJugador: Este frame se mostrará a los demás jugadores de la mesa (informándoles sobre quién es el jefe de mesa en una etiqueta) mientras la mesa no esté completa.
- ConfigFrame: Este frame se mostrará cuando se pulse el botón “Configuración” del MainFrame; en él, se encontrará información sobre el modo de juego y tiempo por turno de la partida (dos radiobuttons), así como si se aceptan o no invitados en la mesa (checkbox). Pulsando “Aceptar Cambios” se sale del frame. Sólo el Jefe de Mesa podrá cambiar la configuración.
- ComienzaPartida: Este frame se mostrará a todos los jugadores una vez la mesa esté completa, con la información sobre la configuración del juego y el jefe de mesa. También aparecerá una cuenta atrás de un minuto (label), que es el tiempo máximo del que dispone un jugador para aceptar (timeout). Pulsando el botón “Aceptar”, se saldrá del frame. Pulsando el botón “Rechazar” o caducando el timeout, se saldrá del frame y se regresará a SalaFrame.
- AceptarPausa: Este frame se le mostrará al Jefe de Mesa con el nombre del jugador que solicita la pausa (botón “Pausar”). Pulsando “Rechazar” se ignorará la pausa y se ocultará el frame. Pulsando “Aceptar” se pausará el juego, mostrándose el frame “PausarInfo” a todos los jugadores.
- PausarInfo: Este frame se mostrará a todos los jugadores cuando el Jefe de Mesa pulse el botón “Pausar” o cuando pulse “Aceptar” en AceptarPausa. En el frame aparecerá una cuenta atrás (label) de un minuto. Pasado este tiempo, se ocultará el frame y se reanudará el juego.
- PartidaTerminada: Este frame se mostrará a todos los jugadores cuando la partida termine; consta de una tabla con los nombres de los jugadores y la puntuación de cada uno en la partida, la puntuación global y el ELO anteriores a la partida (jugadores registrados) y el incremento del ELO (en modo Competición). Se informará del ganador (label) y aparecerá una cuenta atrás de un minuto para aceptar una nueva partida. Pulsando el botón “Aceptar”, se saldrá del frame. Pulsando el botón “Rechazar” o caducando el timeout, se saldrá del frame y se regresará a SalaFrame.
- DeseaAbandonar: Este frame se mostrará cuando un jugador pulse el botón “Abandonar” en el transcurso de una partida. Pulsando “Aceptar” se ocultará el frame y se volverá a SalaFrame. Pulsando “Cancelar” se ocultará el frame.

4.1.3. PROTOCOLO

Para la comunicación entre el Servidor y el Cliente, se ha elaborado un Protocolo sobre TCP (de tipo binario), basado en el envío y recepción de paquetes. Estos paquetes son instancias de clases que implementan el interfaz Paquete.

Como ya se ha comentado, por cada conexión, tanto el Servidor como el Cliente crearán dos hilos: Enviador y Recibidor. Estos hilos estarán continuamente ejecutándose hasta que la conexión se interrumpa. El Enviador se encargará de utilizar el Protocolo para ir enviando los paquetes (de una cola bloqueante); el Recibidor mientras tanto se encargará de la recepción de paquetes por medio del Protocolo y de realizar las acciones asociadas a ellos.

4.1.3.1. Clase Protocolo


 Protocolo
<i>Attributes</i>
<code>public Class tabla[0..*]</code>
<i>Operations</i>
<code>public void enviar(DataOutputStream o, Paquete p, String cliente)</code>
<code>public Paquete recibir(DataInputStream i, String cliente)</code>

Ilustración 4.4 Clase Protocolo

• **Atributos:**

- **tabla:** Se trata de una tabla de elementos que representarán las clases de los diferentes paquetes a implementar.

• **Métodos:**

- **enviar():** El Enviador llamará a este método cada vez que se quiera enviar un paquete (serialización o Marshalling).
- **recibir():** El Recibidor llamará a este método a la espera de la recepción de paquetes. Utilizando la Tabla, se comprobará qué tipo de Paquete es, y se creará una instancia del paquete a la que se le rellenarán los atributos (deserialización).

4.1.3.2. El Interfaz Paquete

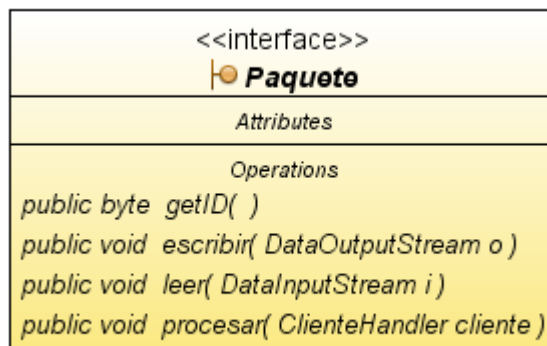


Ilustración 4.5 Interfaz Paquete

• **Atributos:**

- Ninguno.

• **Métodos:**

- **getId():** Este método devolverá un número que identificará de manera única a cada paquete.
- **escribir():** Este método será llamado por el Protocolo para escribir el paquete en el stream de salida del socket.
- **leer():** Este método será llamado por el Protocolo para, una vez creada una instancia del paquete recibido, leer el paquete del stream de entrada del socket.
- **procesar():** En este método se indicará qué se debe hacer cuando se reciba un determinado paquete.

A continuación, se listarán las distintas clases que implementan el interfaz Paquete, y se detallará su funcionalidad.

PaqueteError

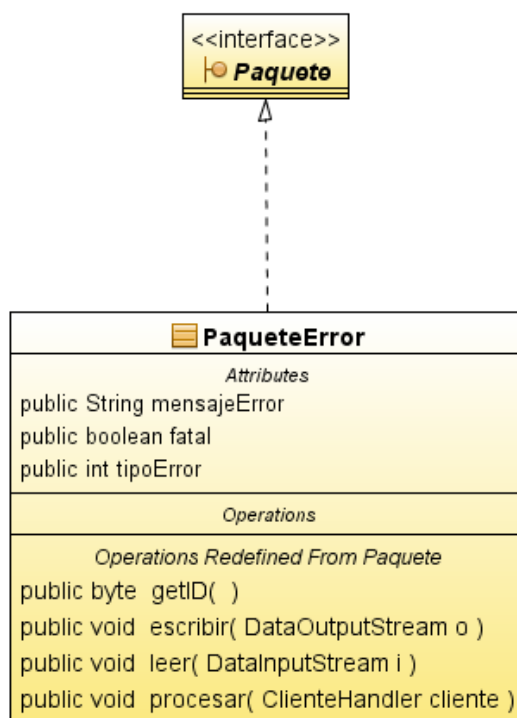


Ilustración 4.6 PaqueteError

Este paquete se enviará del Servidor al Cliente cuando se produzca cualquier tipo de situación anómala; contiene el tipo de error, el mensaje a mostrar y un booleano que indicará si es fatal o no (en cuyo caso el cliente desconectará)

• Servidor:

El Servidor mandará este paquete con el booleano fatal a true:

- Cuando un cliente intente acceder y el sistema se encuentre en su capacidad máxima.
- Cuando un cliente se intente conectar con un nick ya utilizado.
- Cuando no se ha introducido ningún nick
- Cuando un invitado intenta entrar con un nick registrado.
- Cuando un usuario registrado utilice una contraseña errónea.
- Cuando un cliente se intente conectar como usuario registrado con un nick no registrado.

El Servidor mandará este paquete con el booleano fatal a false:

- Cuando un cliente intente acceder a una Mesa llena.
- Cuando un invitado intente acceder a una mesa que no permita invitados.
- Cuando un cliente intente acceder a una Sala llena.

• Cliente:

Procesar: Al recibir el paquete, en el cliente se mostrarán las ventanas de error con el mensaje que viene en mensajeError.

Si el booleano fatal vale true, se procederá a la desconexión (cierre del socket e interrupción de los hilos enviador y receptor).

PaqueteLogin

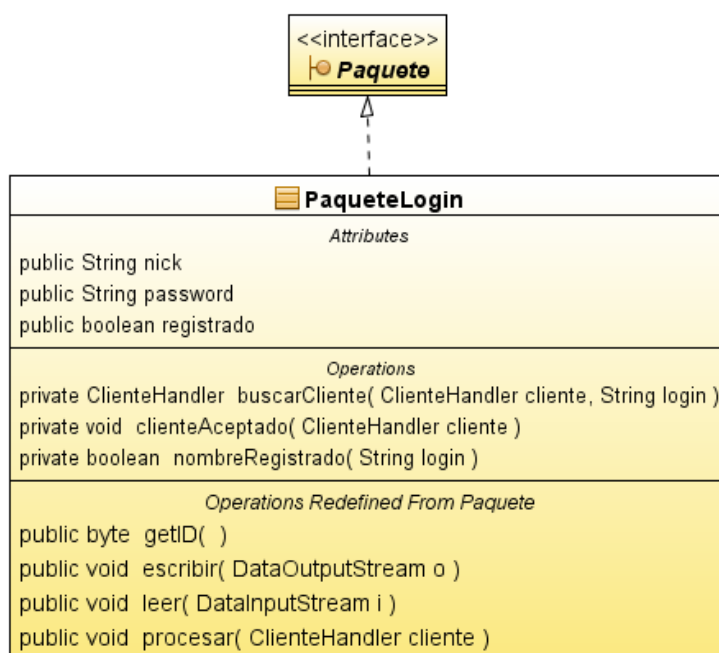


Ilustración 4.7 PaqueteLogin

Este paquete se enviará del Cliente al Servidor para identificarse con sus credenciales.

• Servidor:

Procesar: Al recibir el paquete, en el Servidor se procede a comprobar si se acepta al cliente o no.

- En caso de no aceptación, se enviará al Cliente un paquete (PaqueteError).
- En caso de aceptación, el cliente se añadirá a la lista de clientes del sistema y de jugadores del Salón. Además, se enviará al cliente un paquete PaqueteSalon, y a todos los jugadores del Salón un paquete PaqueteJugadorEntraSalón.

• Cliente:

El Cliente enviará este paquete indicando el nick, si se va a jugar como usuario registrado y, en este caso, la contraseña.

PaqueteSalon

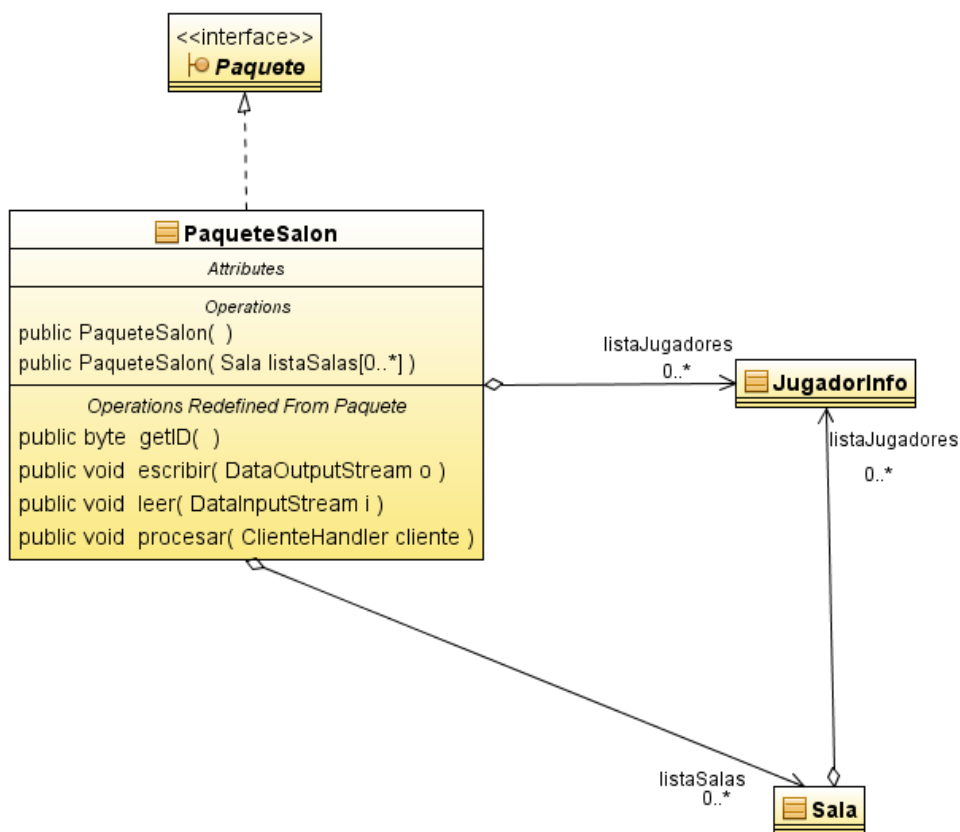


Ilustración 4.8 PaqueteSalon

Este paquete se enviará del Servidor al Cliente para mostrar la información del Salón Principal: la lista de Salas y de jugadores del Salón.

• Servidor:

El Servidor mandará este paquete con la lista de las Salas y de los jugadores del Salón cuando un cliente sea aceptado tras hacer login o cuando vuelva de una Sala al Salón Principal.

• **Ciente:**

Procesar: En el Cliente tras recibir el paquete, se mostrará el frame SalonFrame y se actualizarán las tablas de las Salas y de los jugadores del Salón.

PaqueteSelecSala

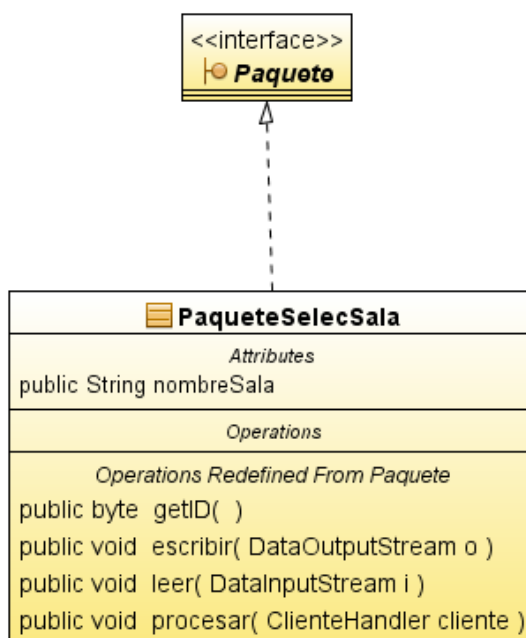


Ilustración 4.9 PaqueteSelecSala

Este paquete se enviará del Cliente al Servidor cuando se seleccione una Sala para acceder a ella.

• **Servidor:**

Procesar: Al recibir este paquete, en el Servidor se comprobará que la sala no está llena; si lo está, se mandará al Cliente un paquete **PaqueteError**. Si no, se eliminará al jugador de la lista de jugadores del Salón, se añadirá a la lista de jugadores de la Sala y se le enviará al cliente un paquete **PaqueteSala**. También se enviará un paquete **PaqueteJugadorEntraSala** a todos los jugadores de la Sala.

• **Ciente:**

Se enviará el paquete con el nombre de la Sala seleccionada cuando el cliente seleccione una Sala y pulse el botón “Entrar” del frame SalonFrame.

PaqueteSala

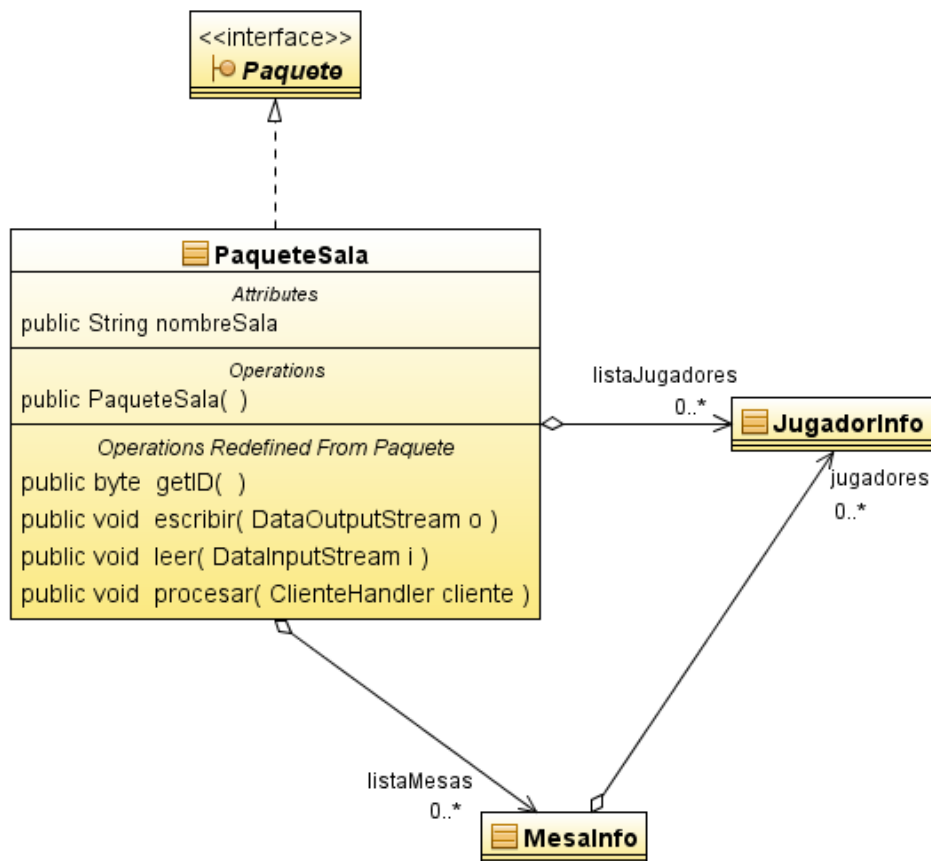


Ilustración 4.10 PaqueteSala

Este paquete se enviará del Servidor al Cliente para mostrar la información de la Sala: la lista de Mesas y de jugadores de la Sala.

• Servidor:

El Servidor mandará este paquete con la lista de las Mesas y de los jugadores de la Sala cuando un cliente entre en ella.

• Cliente:

Procesar: En el Cliente, tras recibir el paquete, se mostrará el frame `SalaFrame` y se actualizarán las tablas de las Mesas y de los jugadores del Sala (así como la tabla de jugadores de cada Mesa cuando se seleccione en la tabla de las Mesas).

PaqueteChatSalon

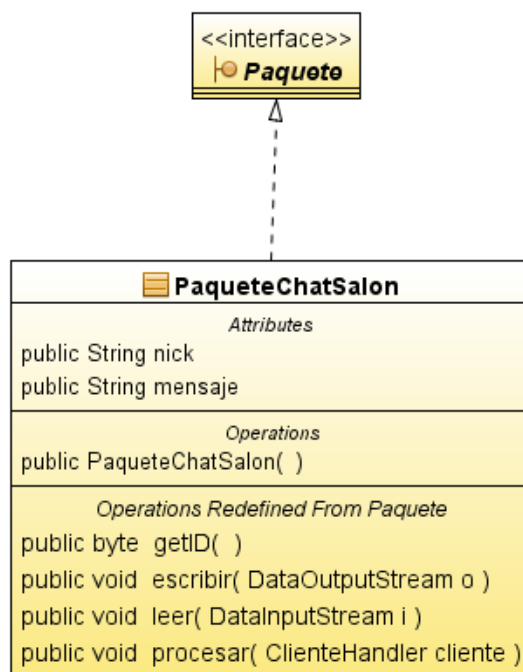


Ilustración 4.11 PaqueteChatSalon

Este paquete se enviará del Cliente al Servidor con el nick del Cliente y el mensaje enviado. Después de esto, también se enviará del Servidor a todos los clientes del Salón.

• Servidor:

Procesar: Al recibir este paquete, en el Servidor se reenviará a todos los jugadores del Salón Principal.

• Cliente:

Se enviará este paquete con el nick del jugador y el mensaje cada vez que el Cliente pulse el botón de chat del Salón o la tecla “Enter”.

Procesar: Al recibir este paquete del Servidor, se mostrará en el chat del frame SalonFrame el nick del jugador y el mensaje que envió.

PaqueteChatSala

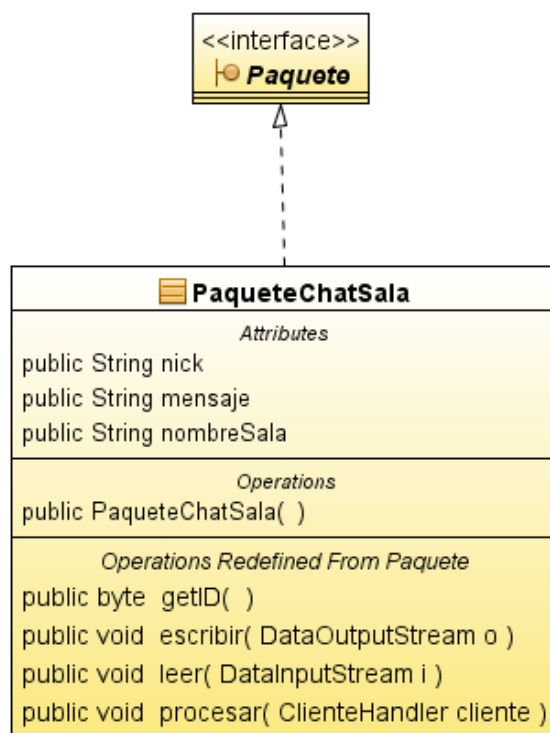


Ilustración 4.12 PaqueteChatSala

Este paquete se enviará del Cliente al Servidor con el nick del Cliente y el mensaje enviado. Después de esto, también se enviará del Servidor a todos los clientes de la Sala.

• Servidor:

Procesar: Al recibir este paquete, en el Servidor se reenviará a todos los jugadores que se encuentren en la misma Sala que el cliente.

• Cliente:

Se enviará este paquete con el nick del jugador y el mensaje cada vez que el Cliente pulse el botón de chat de la Sala o la tecla “Enter”.

Procesar: Al recibir este paquete del Servidor, se mostrará en el chat del frame SalaFrame el nick del jugador y el mensaje que envió.

PaqueteJugadorEntraSalon

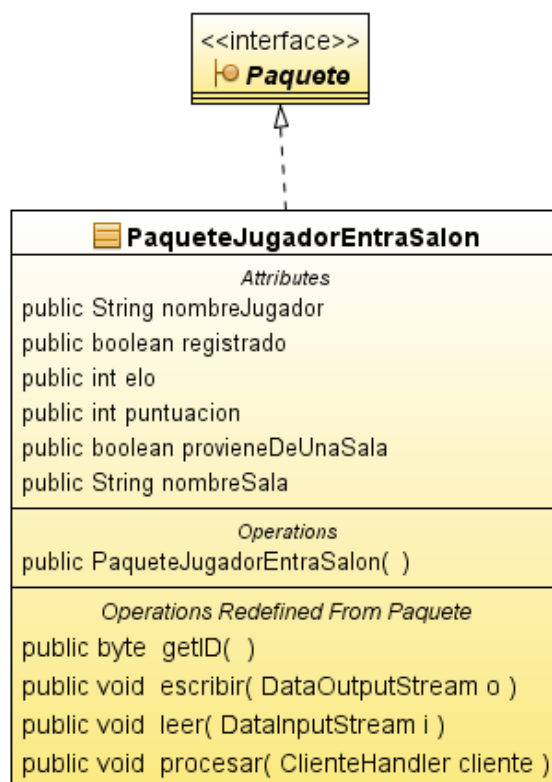


Ilustración 4.13 PaqueteJugadorEntraSalon

Este paquete se enviará del Servidor a todos los Clientes del Salón cada vez que un jugador entre en el Salón Principal.

• Servidor:

El servidor enviará este paquete a todos los jugadores del Salón Principal con los datos del jugador que entra al Salón.

• Cliente:

Procesar: En el cliente al recibir el paquete se mostrará el frame `SalonFrame` y se actualizará la tabla de jugadores del Salón.

PaqueteJugadorEntraSala

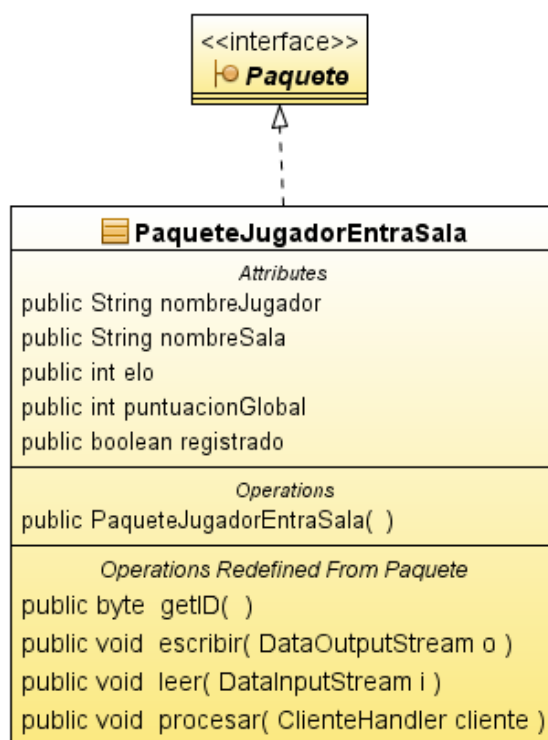


Ilustración 4.14 PaqueteJugadorEntraSala

Este paquete se enviará del Servidor a todos los Clientes de la Sala cada vez que un jugador entre en ella, para informar de esta situación.

• Servidor:

El servidor enviará este paquete a todos los jugadores que se encuentren en la misma sala que el cliente que ha entrado en ella, con los datos del jugador que entra al Salón.

• Cliente:

Procesar: En el cliente al recibir el paquete se mostrará el frame `SalaFrame` y se actualizará la tabla de jugadores de la Sala.

PaqueteVolverSalon

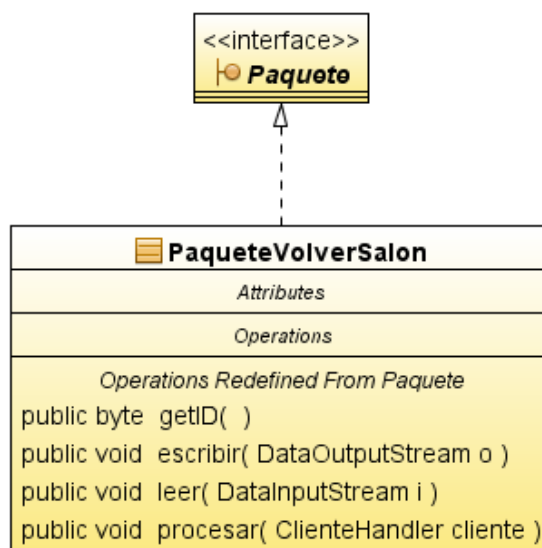


Ilustración 4.15 PaqueteVolverSalon

Este paquete se enviará del Cliente al Servidor cada vez que un jugador desee volver al Salón Principal desde una Sala.

• Servidor:

Procesar: Al recibir este paquete, en el Servidor se eliminará al jugador de la lista de jugadores de la Sala y se añadirá a la lista de jugadores del Salón. Además, se enviará un paquete **PaqueteJugadorEntraSalon** a todos los clientes del Salón Principal, y un paquete **PaqueteJugadorSaleSala** a todos los clientes que se encontraban en la Sala del cliente. Por último, se enviará al cliente un paquete **PaqueteSalon**.

• Cliente:

El cliente enviará este paquete al Servidor cuando pulse el botón “Volver” de la Sala.

PaqueteSaleSala

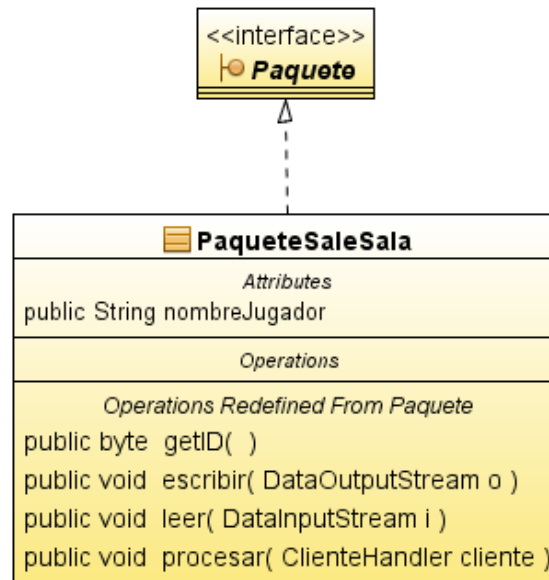


Ilustración 4.16 PaqueteSaleSala

Este paquete se enviará del Servidor a todos los Clientes de la Sala para informar cuando un Cliente ha vuelto al Salón Principal.

• Servidor

El Servidor enviará este paquete a todos los jugadores de la Sala con el nombre del jugador que abandona la Sala.

• Cliente

Procesar: Cuando el Cliente reciba este paquete se borrará al jugador, cuyo nombre viene indicado en el paquete, de la tabla de jugadores de la Sala en el frame SalaFrame.

PaqueteJugadorSeVa

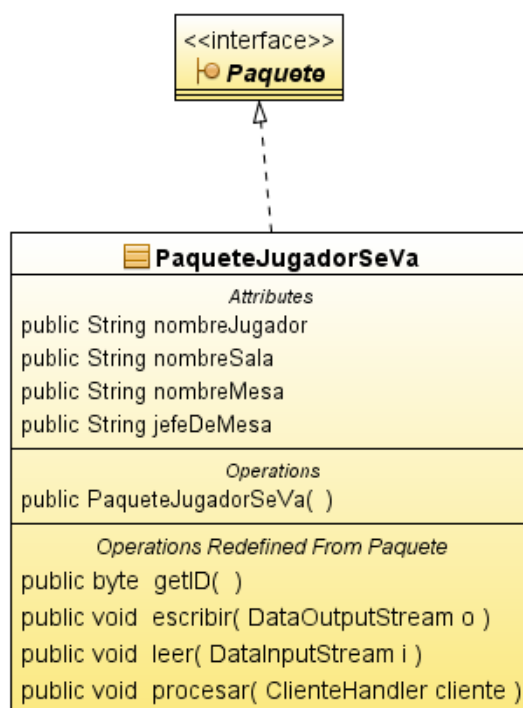


Ilustración 4.17 PaqueteJugadorSeVa

Este paquete se envía del Servidor a los Clientes para informar de la desconexión de un determinado Cliente.

• Servidor:

- Si el Cliente se encontraba en el Salón Principal, se elimina al jugador de la lista de jugadores del Salón y el paquete se envía a todos los jugadores del Salón.
- Si el Cliente se encontraba en una Sala, se elimina al jugador de la lista de jugadores de la Sala y el paquete se envía a todos los jugadores de la Sala.
- Si el Cliente se encontraba en una Mesa, el paquete se envía a todos los jugadores de la Mesa (con el nombre del nuevo Jefe de Mesa) y los de la Sala.

• Cliente:

Procesar: Cuando el Cliente reciba este paquete, se procederá a actualizar las tablas de jugadores dependiendo de la información que le llegue en él. En el caso de estar una partida en curso, se realizarán además las acciones oportunas.

PaqueteSelecMesa

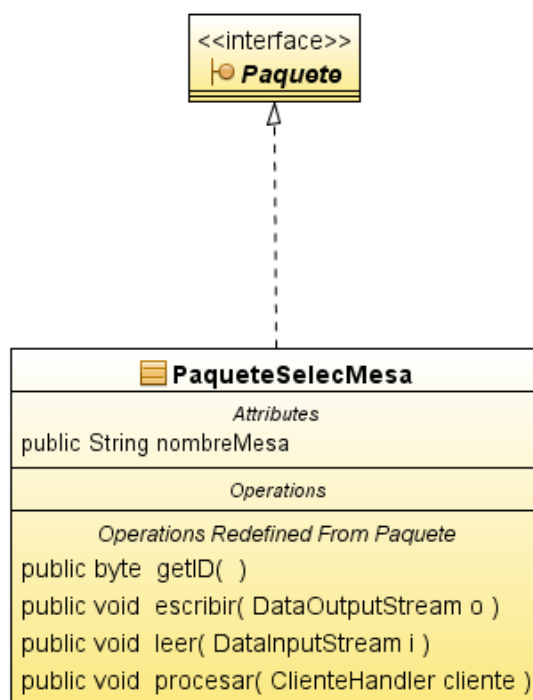


Ilustración 4.18 PaqueteSelecMesa

Este paquete se enviará del Cliente al Servidor cuando se seleccione una Mesa para acceder a ella.

• Servidor:

Procesar: Al recibir este paquete, en el Servidor se comprobará que la mesa no está llena; si lo está, se mandará al Cliente un paquete **PaqueteError**. Si el Cliente que envía el paquete es un invitado y la mesa está configurada para no permitir invitados, se enviará un paquete **PaqueteError**. Si no, se añadirá al jugador a la lista de jugadores de la Mesa y se le enviará al cliente un paquete **PaqueteMesa**. También se enviará un paquete **PaqueteJugadorEntraMesa** a todos los jugadores de la Sala.

Por último, se comprobará si se ha completado el número de jugadores necesarios de la mesa y, en ese caso, se realizarán las acciones para comenzar la partida.

• Cliente:

Se enviará el paquete con el nombre de la Mesa seleccionada cuando el cliente seleccione una Mesa y pulse el botón “Entrar” del frame **SalaFrame**.

PaqueteJugadorEntraMesa

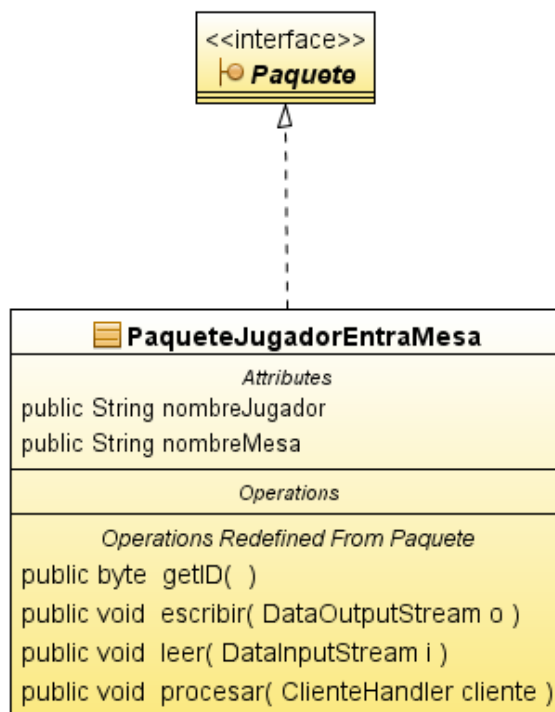


Ilustración 4.19 PaqueteJugadorEntraMesa

Este paquete se enviará del Servidor a los Clientes de la Sala para informar cuando un jugador entre en una mesa.

• Servidor:

Este paquete se enviará cuando un jugador entre a una Mesa, con el nombre del jugador y el de la Mesa.

• Cliente:

Procesar: Tras recibir el paquete, en el cliente se actualizarán las tablas de información de las mesas de la Sala y, para los jugadores que se encuentren en la misma mesa, la tabla de jugadores de la mesa.

PaqueteMesa

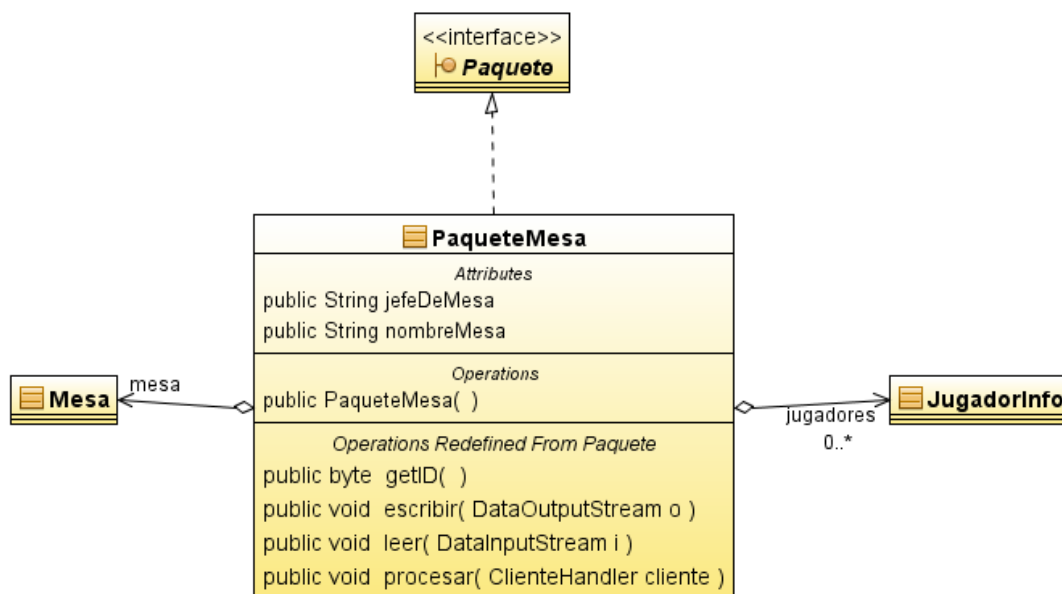


Ilustración 4.20 PaqueteMesa

Este paquete se enviará del Servidor al Cliente para mostrar la información de la Mesa: la lista de jugadores de la Sala, el nombre del Jefe de Mesa y el estado de la mesa.

• Servidor:

El Servidor mandará este paquete con la lista de los jugadores de la Mesa, el nombre del Jefe de Mesa y el estado de la mesa (tablero) cuando un cliente entre en ella.

• Cliente:

Procesar: En el Cliente, tras recibir el paquete, se mostrará el frame de la Mesa y se actualizarán la tabla de los jugadores de la Mesa, el tablero y el nombre del Jefe de Mesa; también se mostrarán los distintos frames de bienvenida.

PaqueteConfig

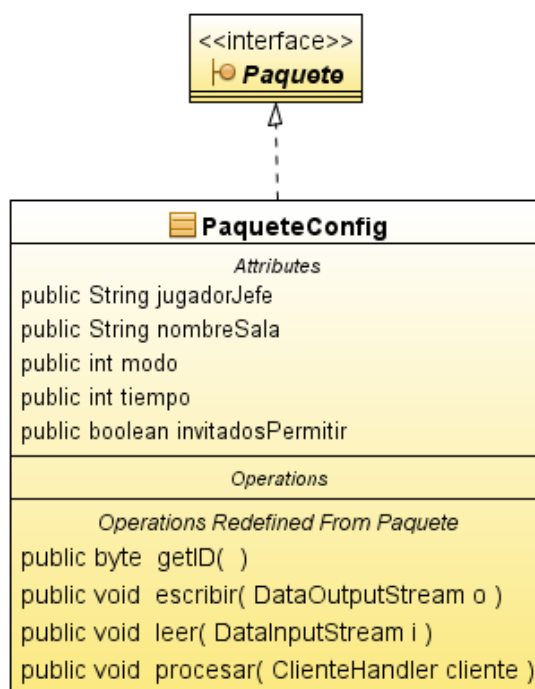


Ilustración 4.21 PaqueteConfig

Este paquete se enviará del Cliente al Servidor con el propósito de cambiar la configuración de la mesa.

- **Servidor:**

Procesar: Al recibir este paquete, en el Servidor se comprobará si el paquete lo ha enviado el Jefe de Mesa antes del comienzo de la partida; en ese caso, se aplicarán los cambios en la configuración de la partida.

- **Cliente:**

El cliente enviará este paquete al Servidor cuando pulse el botón “Aceptar Cambios” de la Mesa, con los datos de la configuración.

PaqueteSalirMesa

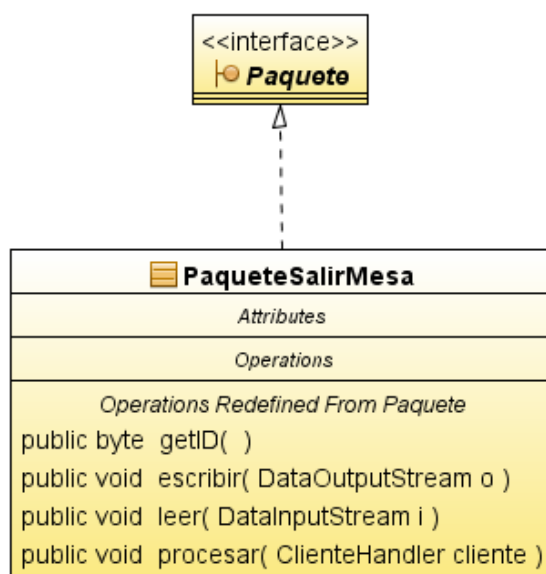


Ilustración 4.22 PaqueteSalirMesa

Este paquete se envía del Cliente al Servidor, cada vez que el Cliente desee abandonar la Mesa en la que se encuentre.

• Servidor:

Procesar: Al recibir este paquete, en el Servidor se eliminará al cliente de la lista de jugadores de la Mesa y se le enviará un paquete **PaqueteSala**. Se enviará un paquete **PaqueteSaleMesa** a todos los jugadores de la Sala. En caso de estar una partida en curso, se realizarán las acciones convenientes.

• Cliente:

El Cliente enviará este paquete al Servidor cuando pulse el botón “Abandonar” de la Mesa cuando la partida aún no haya comenzado o el botón “Aceptar” del frame **DeseaAbandonar** cuando la partida esté en curso.

PaqueteSaleMesa

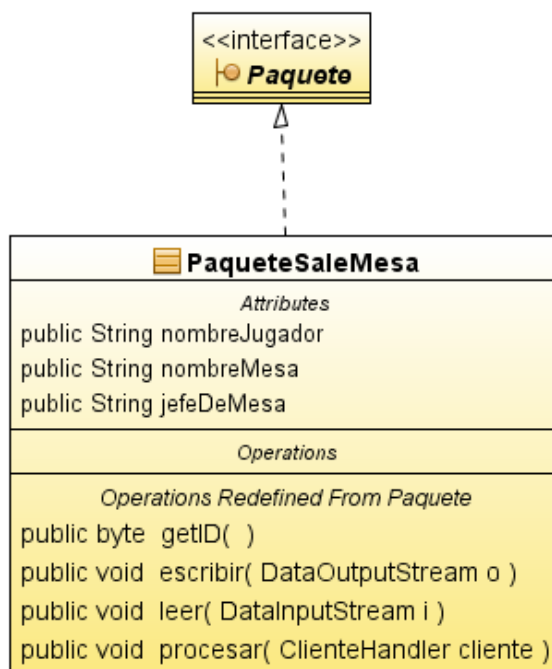


Ilustración 4.23 PaqueteSaleMesa

Este paquete se enviará del Servidor a todos los Clientes de la Sala para informar de que un Cliente ha abandonado una mesa.

• Servidor:

El Servidor enviará este paquete con el nombre del jugador que abandona y el nombre de la mesa a todos los Clientes de la Sala. También se enviará el nombre del nuevo Jefe de Mesa.

• Cliente:

Procesar: En el Cliente, tras recibir el paquete se actualizarán las listas de jugadores de la Mesa y para los jugadores que se encuentren en la mesa, el nombre del Jefe de Mesa. Además si la partida estaba en curso, se realizarán las acciones oportunas.

PaqueteChatMesa

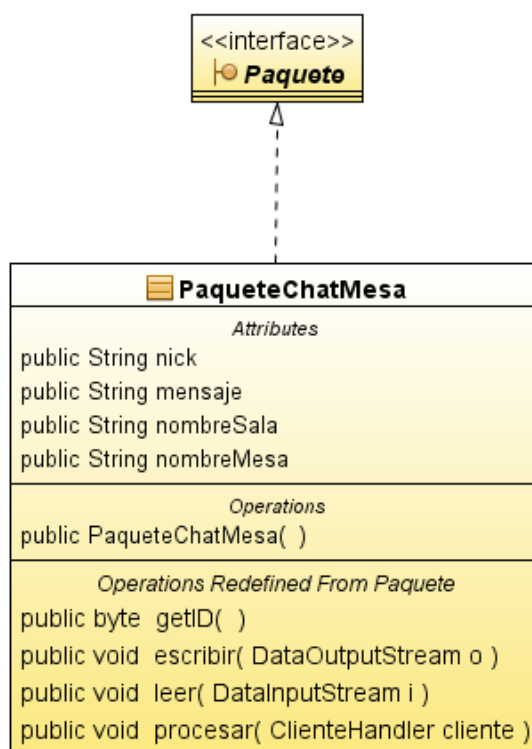


Ilustración 4.24 PaqueteChatMesa

Este paquete se enviará del Cliente al Servidor con el nick del Cliente y el mensaje enviado. Después de esto, también se enviará del Servidor a los Clientes de la mesa si la partida ha comenzado y si no, a todos los Clientes de la Sala.

• Servidor:

Procesar: Al recibir este paquete, en el Servidor se reenviará sólo a los jugadores de la Mesa si la partida está en curso y, a todos los jugadores que se encuentren en la misma Sala que el cliente, si la partida aún no ha comenzado.

• Cliente:

Se enviará este paquete con el nick del jugador y el mensaje cada vez que el Cliente pulse el botón de chat de la Mesa o la tecla “Enter”.

Procesar: Al recibir este paquete del Servidor, se mostrará en el chat de la Mesa o de la Sala, según el caso, el nick del jugador y el mensaje que envió.

PaqueteActualizacion

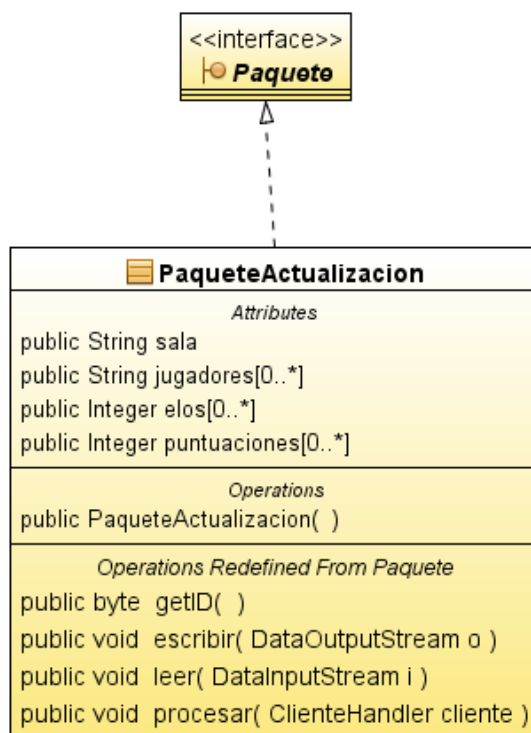


Ilustración 4.25 PaquetActualizacion

Este paquete se enviará del Servidor a todos los Clientes de la Sala para actualizar la información sobre los ELO y Puntuaciones Globales de los jugadores una vez acabada la partida.

• Servidor:

Se enviará un paquete con las listas de los jugadores de la mesa, de las puntuaciones globales y de los ELOs cuando termine una partida.

• Cliente:

Procesar: En el Cliente, tras recibir el paquete, se actualizará la tabla de los jugadores de la Sala con la nueva información.

PaqueteAceptarConfig

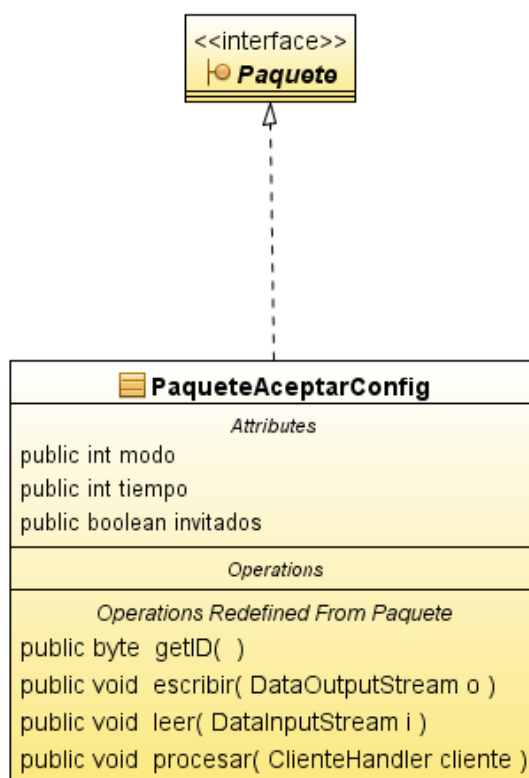


Ilustración 4.26 PaqueteAceptarConfig

Este paquete se enviará a los jugadores de una mesa antes de comenzar una partida con la configuración del juego.

• Servidor:

Se enviará este paquete con los datos de la configuración del juego a los jugadores de una mesa cuando se haya completado el número de jugadores necesarios.

• Cliente:

Procesar: En el Cliente, tras recibir el paquete, se mostrará un frame con la configuración del juego que podrá ser aceptada o rechazada.

PaqueteConfigAceptada

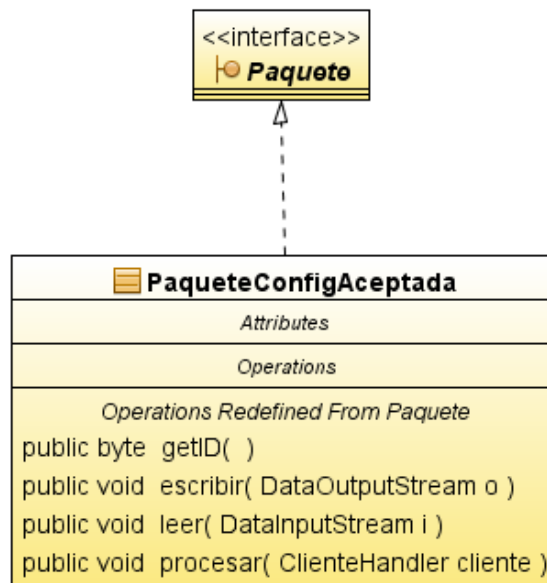


Ilustración 4.27 PaqueteConfigAceptada

Este paquete se enviará del Cliente al Servidor para indicar que se acepta la configuración del juego.

• Servidor:

Procesar: Al recibir este paquete, en el Servidor se llevará una cuenta de los jugadores que aceptan la configuración. Si antes de agotarse el timeout todos los jugadores han aceptado, se procederá al comienzo de la partida, enviándose los paquete **PaqueteComenzarPartida** y **PaqueteTurno** a los clientes de la mesa.

• Cliente:

Se enviará este paquete cuando el cliente pulse “Aceptar” en el frame de aceptación de la Configuración.

PaqueteComenzarPartida

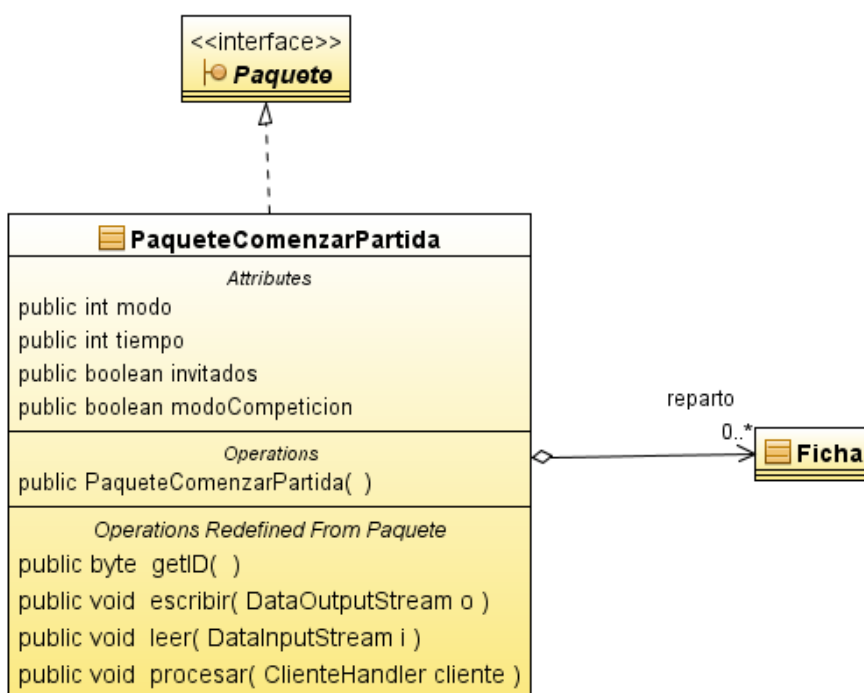


Ilustración 4.28 PaqueteComenzarPartida

Se enviará este paquete del Servidor a los Clientes de la mesa para comenzar la partida.

• Servidor:

El Servidor enviará este paquete con los datos de la configuración, el modo de juego y el reparto inicial correspondiente, a cada uno de los jugadores de la mesa.

• Cliente:

Procesar: En el Cliente, tras recibir este paquete, se actualizará la información de la mesa y el reparto del jugador.

PaquetePausarJuego

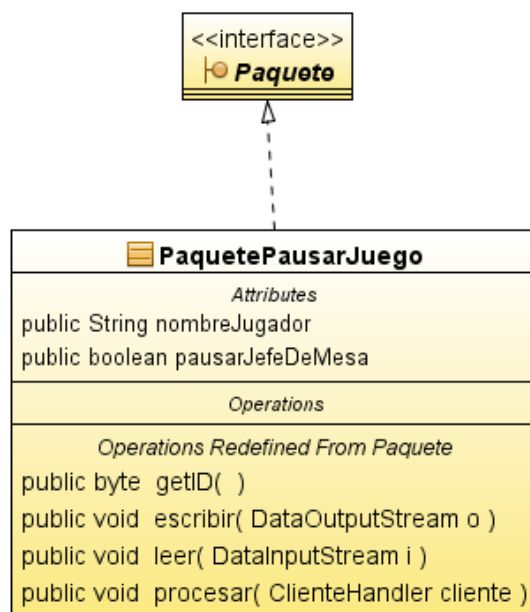


Ilustración 4.29 PaquetePausarJuego

La Pausa sólo podrá ser accionada a través del Jefe de Mesa: de esta forma, este paquete se enviará:

- Si lo envía el Jefe de Mesa, del Cliente al Servidor, y después del Servidor a todos Clientes jugadores de la Mesa.
- Si lo envía un jugador normal, del Cliente al Servidor, del Servidor al Cliente Jefe de Mesa; y si este acepta la solicitud de pausa, nuevamente del Cliente al Servidor.

• Servidor:

Procesar: Si el paquete lo envía el Jefe de Mesa (ya sea porque ha pausado él o porque ha aceptado la solicitud de pausa), se pausará el juego y se reenviará este paquete para notificar la pausa a todos los jugadores de la mesa. Si el paquete lo envía un jugador que no es el Jefe, se reenviará este paquete al Jefe de Mesa para que acepte o no la solicitud de pausa.

• Cliente:

Cuando un jugador pulsa el botón de Pausa de la mesa, se enviará este paquete al Servidor con el nombre del jugador que solicita la pausa.

Procesar:

- Si el Cliente no es el Jefe de Mesa, este paquete sólo le podrá llegar una vez se ha aceptado la pausa; se bloquearán los botones de juego, el tablero y el reparto, y se mostrará un frame con una cuenta atrás.
- Si el Cliente es el Jefe de Mesa, este paquete le puede llegar:
 - 1) para aceptar o no la solicitud de pausa; en este caso, se le mostrará un frame para aceptar o no la solicitud, y si la acepta, se reenviará el paquete al Servidor indicando que se ha aceptado.
 - 2) porque la pausa ha sido aceptada; en este caso, se bloquearán los botones de juego, el tablero y el reparto, y se mostrará un frame con una cuenta atrás.

PaqueteReanudarJuego

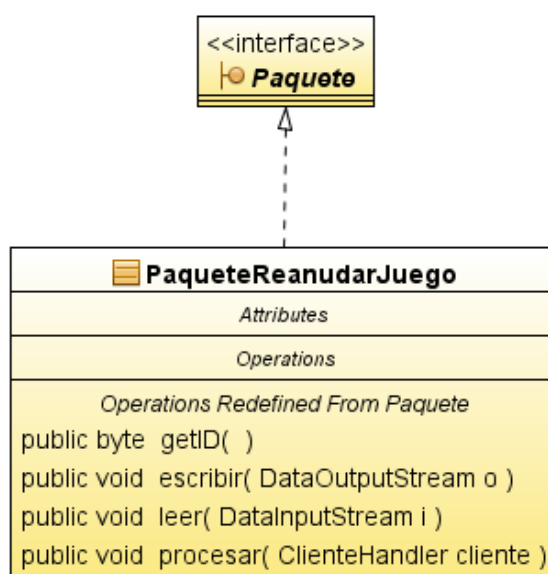


Ilustración 4.30 PaqueteReanudarJuego

Este paquete se enviará del Servidor a los Clientes de la mesa para informarles del fin de la pausa y de la reanudación del juego.

• Servidor:

Este paquete se enviará cuando se agote el tiempo de la Pausa, a todos los jugadores de la mesa.

• Cliente:

Procesar: En el Cliente, tras recibir este paquete, se indicará que el juego se ha reanudado y se desbloquearán el tablero, el reparto y los botones de juego del jugador al que corresponde el turno actual.

PaqueteJugadorMueveFicha

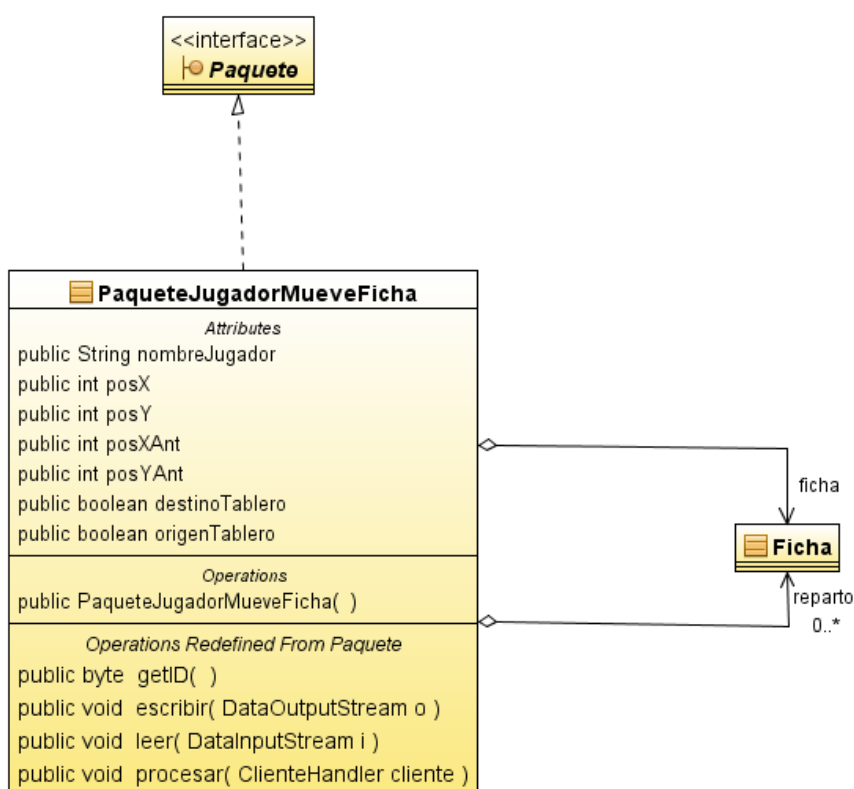


Ilustración 4.31 PaqueteJugadorMueveFicha

Este paquete se enviará del Cliente al Servidor cada vez que mueve una ficha; y del Servidor a los demás Clientes de la Mesa, para informarles del movimiento.

• Servidor:

Procesar: En el Servidor, tras recibir este paquete, se actualizará la información de la mesa, y se reenviará este paquete a los demás jugadores de la mesa.

• Cliente:

Se enviará este paquete cada vez que un jugador mueva una ficha, con la información de la ficha y su posición.

Procesar: Tras recibir este paquete, el Cliente fija la ficha en la posición indicada.

PaqueteTurno

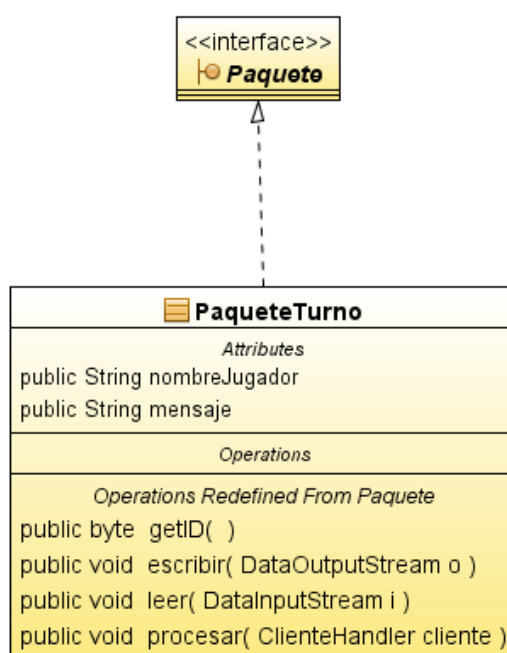


Ilustración 4.32 PaqueteTurno

Este paquete se envía del Servidor todos los Clientes de la mesa cada vez que se produce un cambio de turno.

• Servidor:

Se enviará este paquete con el nombre del jugador al que le corresponde el turno actual, a todos los jugadores de la mesa.

• Cliente:

Procesar: Al recibir este paquete el Cliente, se comprobará si el turno le corresponde y, en este caso, se desbloquearán el tablero, el reparto y los botones del juego.

PaqueteJugadorEnvia

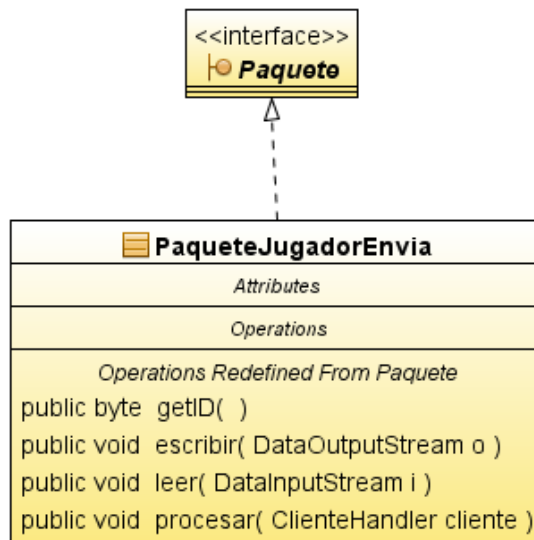


Ilustración 4.33 PaqueteJugadorEnvia

Este paquete se envía del Cliente al Servidor, cada vez que se deseen enviar una jugada.

• Servidor:

Procesar: Al recibir este paquete, en el Servidor se comprobará la jugada y se realizarán las acciones oportunas.

• Cliente:

Este paquete se enviará cuando el Cliente pulse el botón “Enviar” de la Mesa y las fichas estén correctamente colocadas sobre el tablero.

PaqueteJugarFichas

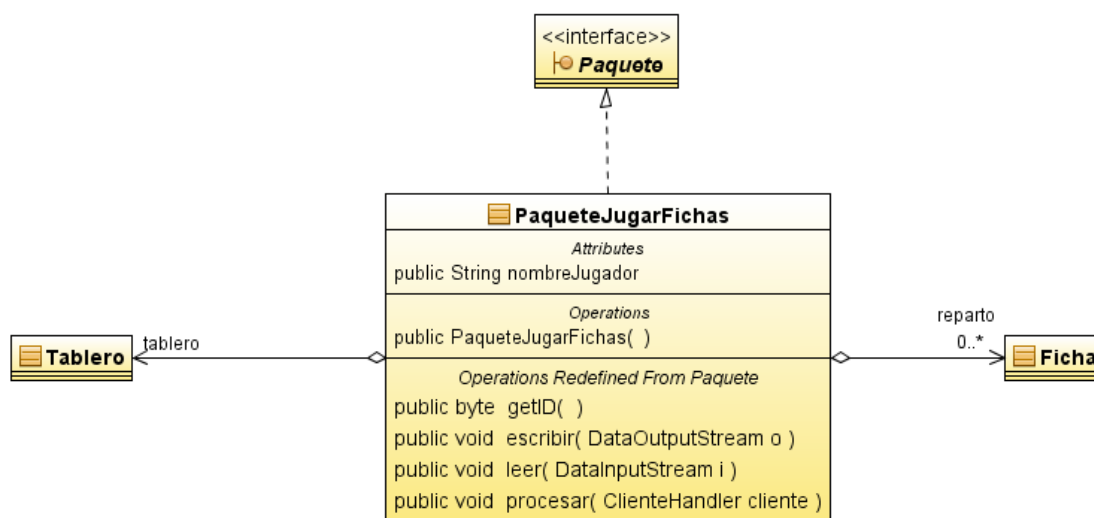


Ilustración 4.34 PaqueteJugarFichas

Este paquete se envía del Servidor al Cliente para informar al Cliente del estado del tablero y del reparto si fuera necesario.

• Servidor:

El Servidor enviará este paquete a todos los jugadores de la mesa en dos ocasiones: cuando dé comienzo la partida y cada vez que se realice una jugada y el correspondiente cambio de turno.

• Cliente:

Procesar: Tras recibir el paquete, el Cliente actualizará el tablero y el reparto si fuera necesario.

PaqueteInfoJugada

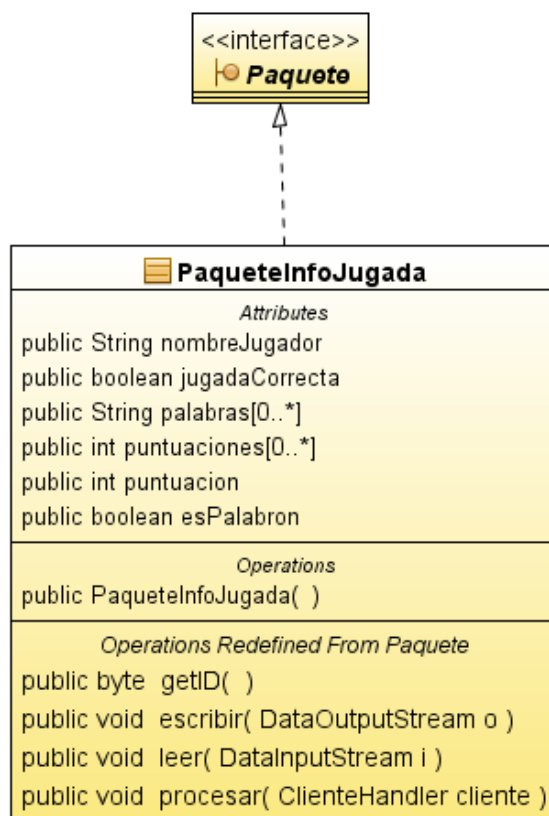


Ilustración 4.35 PaqueteInfoJugada

Este paquete se enviará del Servidor a todos los Clientes de la mesa con la información de una determinada jugada.

• Servidor:

El Servidor enviará este paquete cada vez que se compruebe si una jugada ha sido correcta o no, con los datos de la jugada, a todos los jugadores de la mesa.

• Cliente:

Procesar: En el Cliente, tras recibir este paquete, se mostrará la información sobre la última jugada realizada y se actualizarán las puntuaciones correspondientes.

PaqueteCambiar

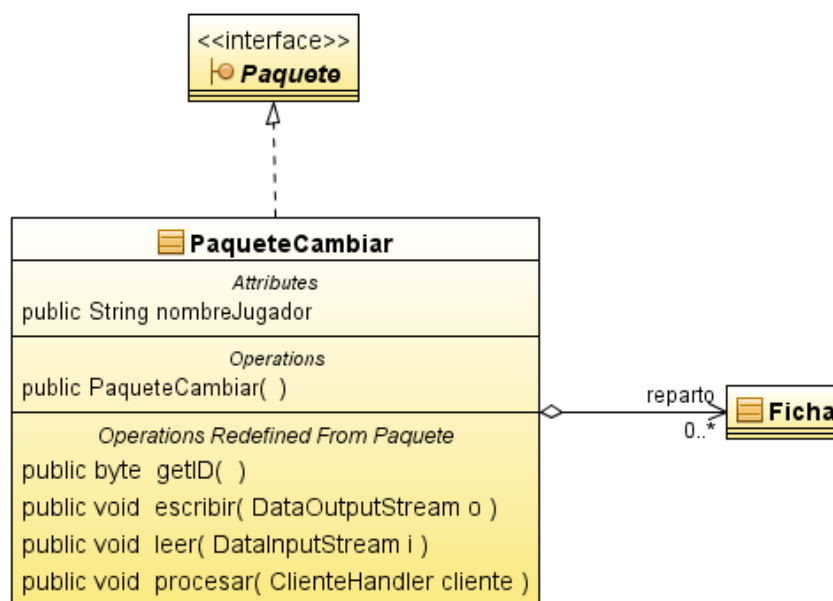


Ilustración 4.36 PaqueteCambiar

Este paquete se envía:

- Del Cliente al Servidor, cuando el Cliente desea cambiar las fichas seleccionadas de su reparto.
- Del Servidor al Cliente, tras haber realizado una jugada correcta para enviarle su nuevo reparto o cuando el Cliente ha solicitado cambiar su reparto.

• Servidor:

Procesar: Cuando el Servidor recibe este paquete, procede a cambiar las fichas seleccionadas de su reparto. Se enviará al Cliente un paquete **PaqueteCambiar** con el nuevo reparto.

• Cliente:

El Cliente enviará este paquete con su reparto de fichas cuando haya seleccionado las fichas que desee cambiar y pulse el botón “Cambiar”.

Procesar: Cuando el Cliente recibe este paquete, procede a cambiar su reparto por el reparto del paquete.

PaquetePasar

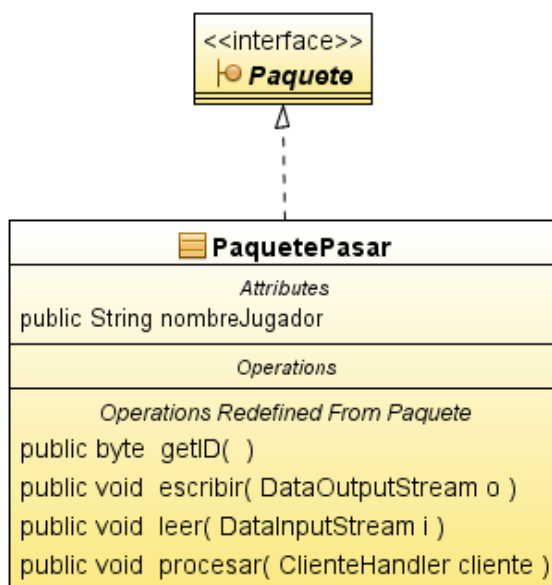


Ilustración 4.37 PaquetePasar

Este paquete se envía del Cliente al Servidor, cada vez que el jugador desee pasar de turno.

• Servidor:

Procesar: Tras recibir el paquete, se procederá al cambio de turno y se enviará a todos los clientes de la mesa un paquete **PaqueteTurno**.

• Cliente:

El cliente enviará este paquete cuando pulse el botón “Pasar” de la mesa.

PaqueteCambiaComodin

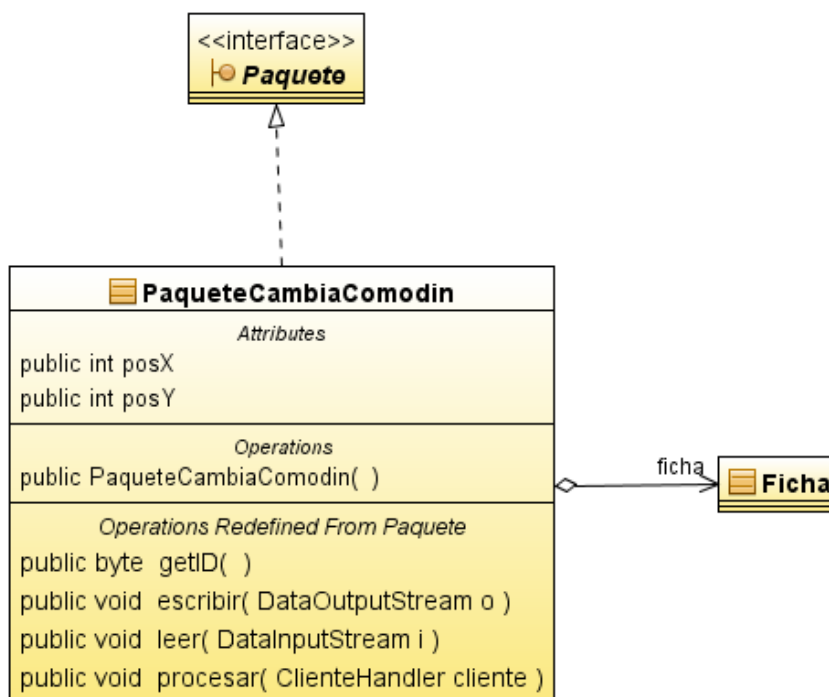


Ilustración 4.38 PaqueteCambiaComodin

Este paquete se enviará del Cliente al Servidor cuando desee utilizar la ficha comodín; y del Servidor a los Clientes de la mesa, para informarles del cambio.

• Servidor:

Procesar: Cuando el Servidor reciba este paquete, éste se reenviará con la ficha y su posición en el tablero a todos los jugadores de la mesa.

• Cliente:

El cliente enviará este paquete cuando, tras colocar la ficha en el tablero y pulsar con el botón derecho sobre ella, seleccione la letra que desea utilizar.

Procesar: Tras recibir este paquete, el Cliente procederá a actualizar la ficha del tablero con la letra elegida.

PaqueteAcabarPartida

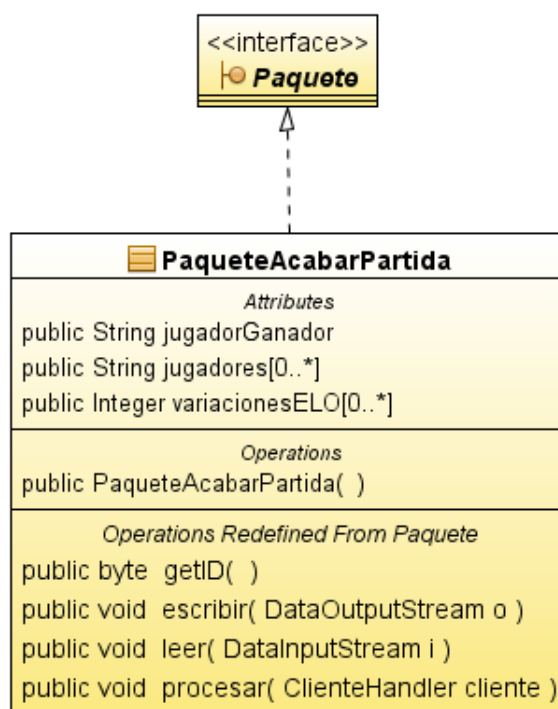


Ilustración 4.39 PaqueteAcabarPartida

Este paquete se enviará del Servidor a los Clientes de la mesa cuando finalice la partida, para informarles del resultado de la misma.

• Servidor:

El paquete se enviará a todos los jugadores de la mesa con el nombre del jugador ganador, las puntuaciones y las variaciones de ELO si la partida se ha disputado en Modo Competición.

• Cliente:

Procesar: En el Cliente, tras recibir este paquete, se actualizará la tabla del final de la partida con los datos recibidos. Además, se parará el tiempo de la partida y se mostrará una cuenta atrás de un minuto, para informar del tiempo del que dispone el jugador para aceptar una nueva partida.

4.2. DIAGRAMAS DE SECUENCIA

En esta sección, desarrollaremos los diagramas de secuencia de los casos más interesantes.

4.2.1. Diagrama de Secuencia: Navegación

En el siguiente diagrama, se mostrará cómo se comporta la aplicación del juego según los distintos escenarios. Para ello, se dispone de dos usuarios que establecen una conexión con el Servidor; en el diagrama, se observa que:

- Los usuarios que están en el Salón Principal, son informados de los nuevos usuarios que acceden al Sistema.
- Los usuarios que están en el Salón Principal, pueden comunicarse entre ellos mediante el Chat.
- Los usuarios que están en una determinada Sala, son informados de los nuevos usuarios que acceden a esa misma Sala (pero no de los que entran a una Sala distinta).
- Los usuarios que están en una determinada Sala, pueden comunicarse entre ellos mediante el Chat y no pueden hacerlo con los usuarios que se encuentren en el Salón Principal o en otras Salas.
- Los usuarios que están en una determinada Sala, pueden regresar al Salón Principal y, de ahí, entrar nuevamente a la Sala que elijan.
- Los usuarios que entran a una Mesa de una Sala, pueden comunicarse con el resto de usuarios de esa misma Sala mientras la partida no haya comenzado.
- Los usuarios que están en una Mesa de una Sala, sólo podrán comunicarse con el resto de usuarios que están en su misma Mesa una vez que la partida haya comenzado.

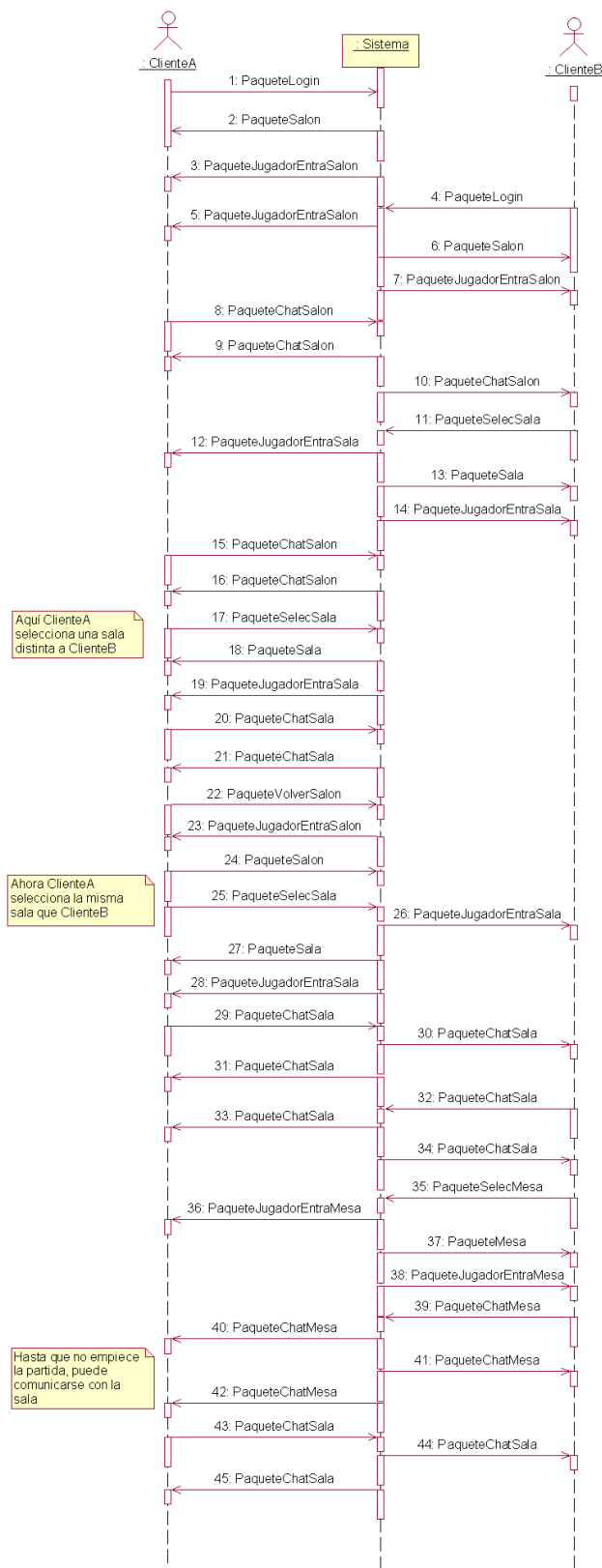


Ilustración 4.40 Diagrama de Secuencia Navegación

En el caso de que un jugador abandone el sistema, se informará de esto de forma similar a lo comentado anteriormente.

4.2.2. Diagrama de Secuencia: Entrar Mesa y Comienzo Partida

Se parte del caso en que ambos usuarios están en la misma Sala, sin haber entrado aún en ninguna Mesa:

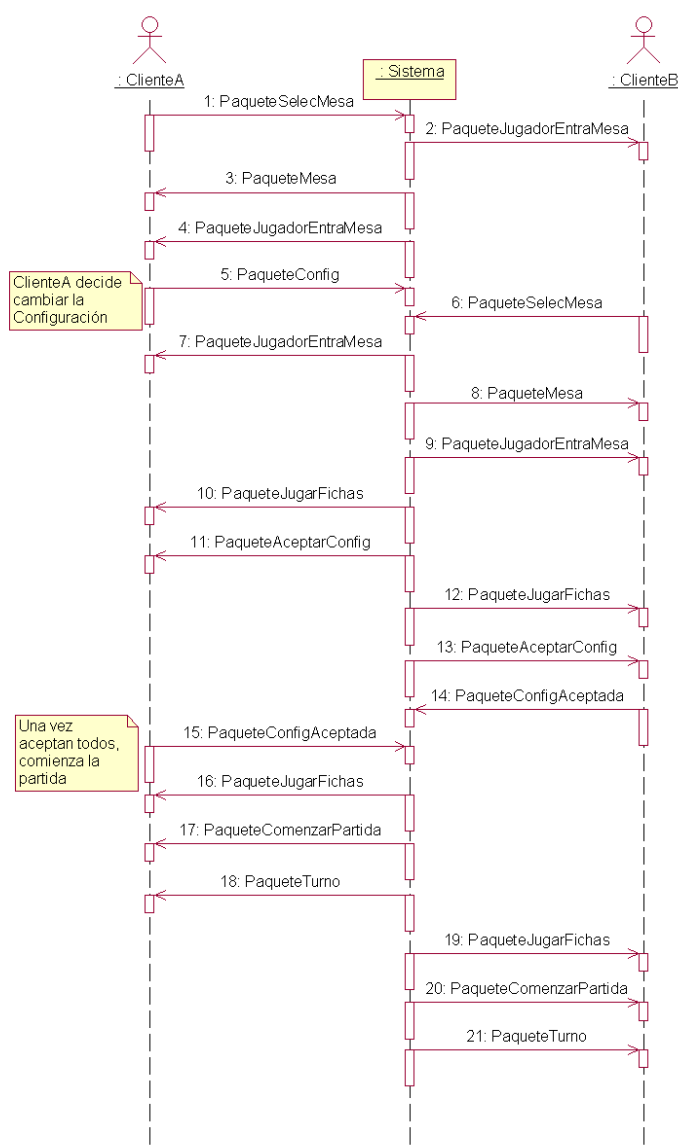


Ilustración 4.41 Diagrama de Secuencia Entrar Mesa y Comienzo Partida

4.2.3. Diagrama de Secuencia: Jugar Partida

Se continúa con el caso anterior: habiendo entrado los usuarios en la mesa, aceptada la configuración por parte de ambos y mandados los primeros paquetes de la partida, se verá ahora parte del desarrollo de la misma. El primer turno corresponde a ClienteA, dado que entró a la mesa antes que ClienteB.

En el diagrama se puede ver qué ocurre cuando se realizan las tres acciones principales de la partida: Enviar, Cambiar y Pasar.

- **Enviar:** En la parte del cliente, se comprueba que las fichas están correctamente colocadas antes de enviar el paquete. Una vez recibe el paquete el servidor, comprobará que la/s palabra/s formada/s son correctas: en caso afirmativo, se enviará un paquete con las nuevas fichas asignadas al usuario que ha realizado la jugada. En ambos casos, se enviará a todos los jugadores un paquete sobre la información de la jugada (si ha sido correcta o no, palabras erróneas, puntos por palabras, puntuación total...), un paquete para el cambio de turno y un paquete para mostrar la actualización del tablero.
- **Cambiar:** El usuario selecciona las fichas que desea cambiar y una vez pulsado el botón, se envía un paquete al servidor. Éste realizará los cambios de las fichas del jugador y le enviará un paquete con su nuevo reparto. Se mandará también un paquete para mostrar la actualización del tablero a todos los jugadores de la mesa.
- **Pasar:** El usuario indicará que desea pasar turno y se enviará un paquete al servidor para que se realice el cambio. El Servidor mandará nuevamente un paquete indicando los nuevos turnos y la actualización del tablero.

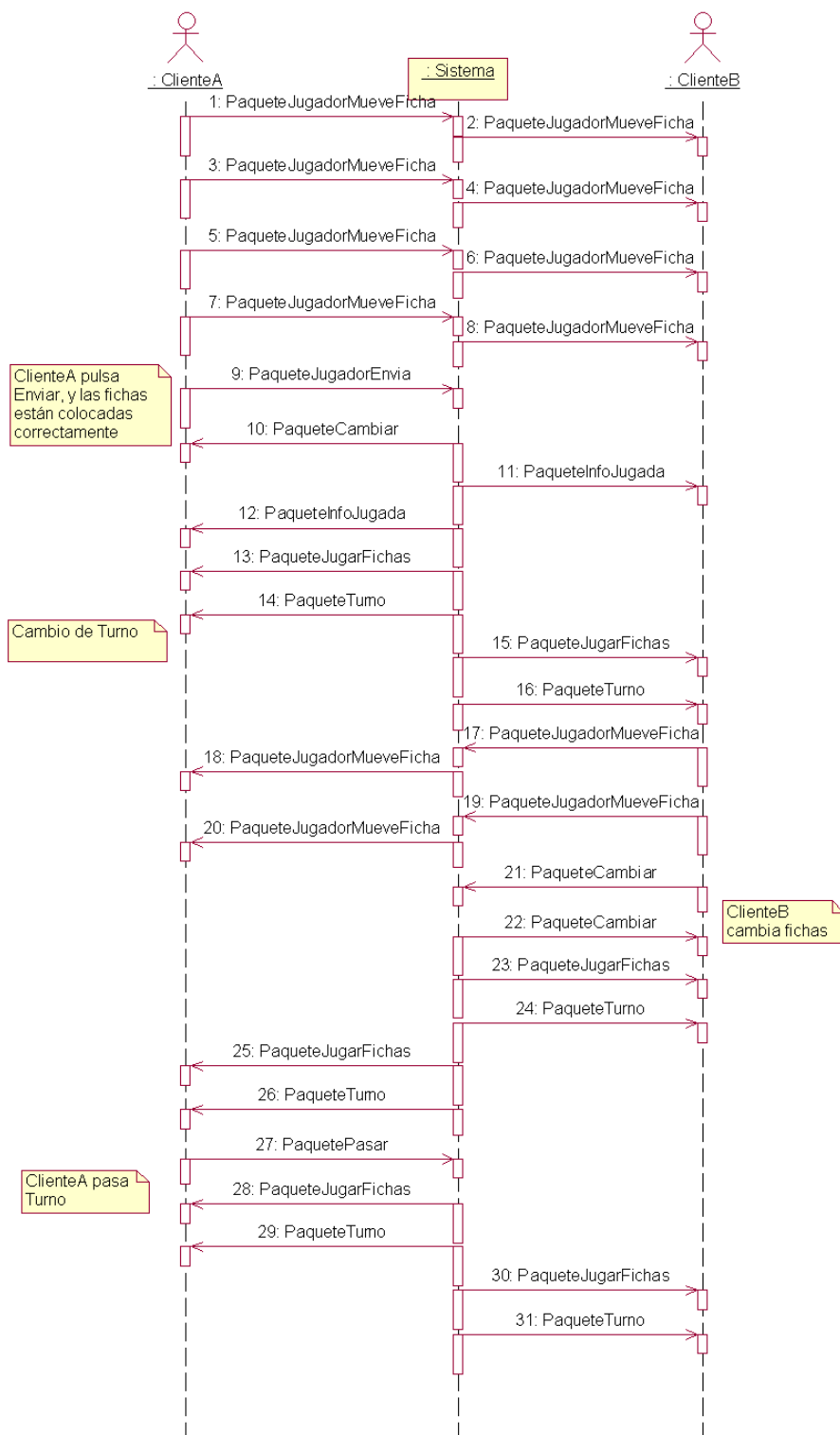


Ilustración 4.42 Diagrama de Secuencia Jugar Partida

4.2.4. Diagrama de Secuencia: Cambiar Comodín

En este apartado se verá qué ocurre cuando un jugador decide utilizar la ficha comodín. Se parte de la situación en que es el turno del ClienteA.

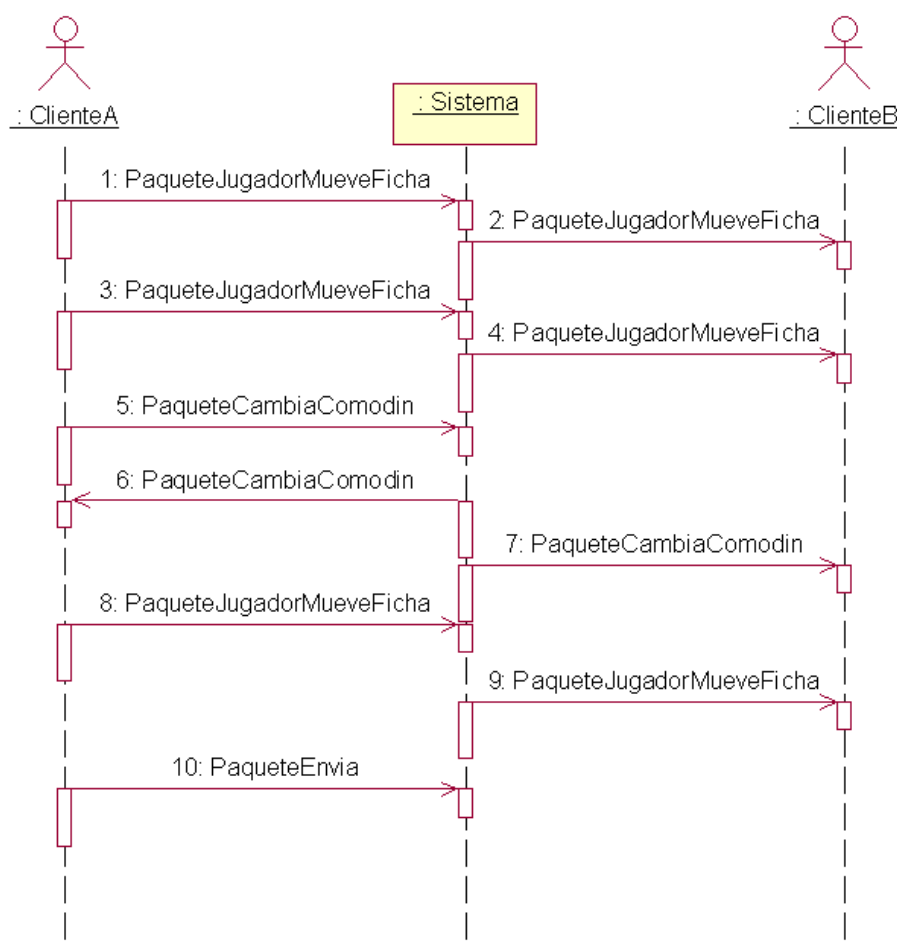


Ilustración 4.43 Diagrama de Secuencia Cambiar Comodín

Para poder cambiar la ficha comodín, debe ser colocada sobre el tablero; una vez hecho esto, el jugador elegirá la letra que desea utilizar. Al hacerlo, se mandará un paquete al servidor y, éste a su vez, mandará un paquete a todos los jugadores de la mesa informándoles de la actualización.

4.2.5. Diagrama de Secuencia: Acabar Partida

Al acabar una partida, se preguntará a los jugadores si desean jugar una nueva partida.

Se parte de la situación de un jugador que, al realizar una jugada, consigue alcanzar el número de puntos requeridos para terminar.

En el primer diagrama, se muestra el caso en que ambos jugadores deciden jugar una nueva partida.

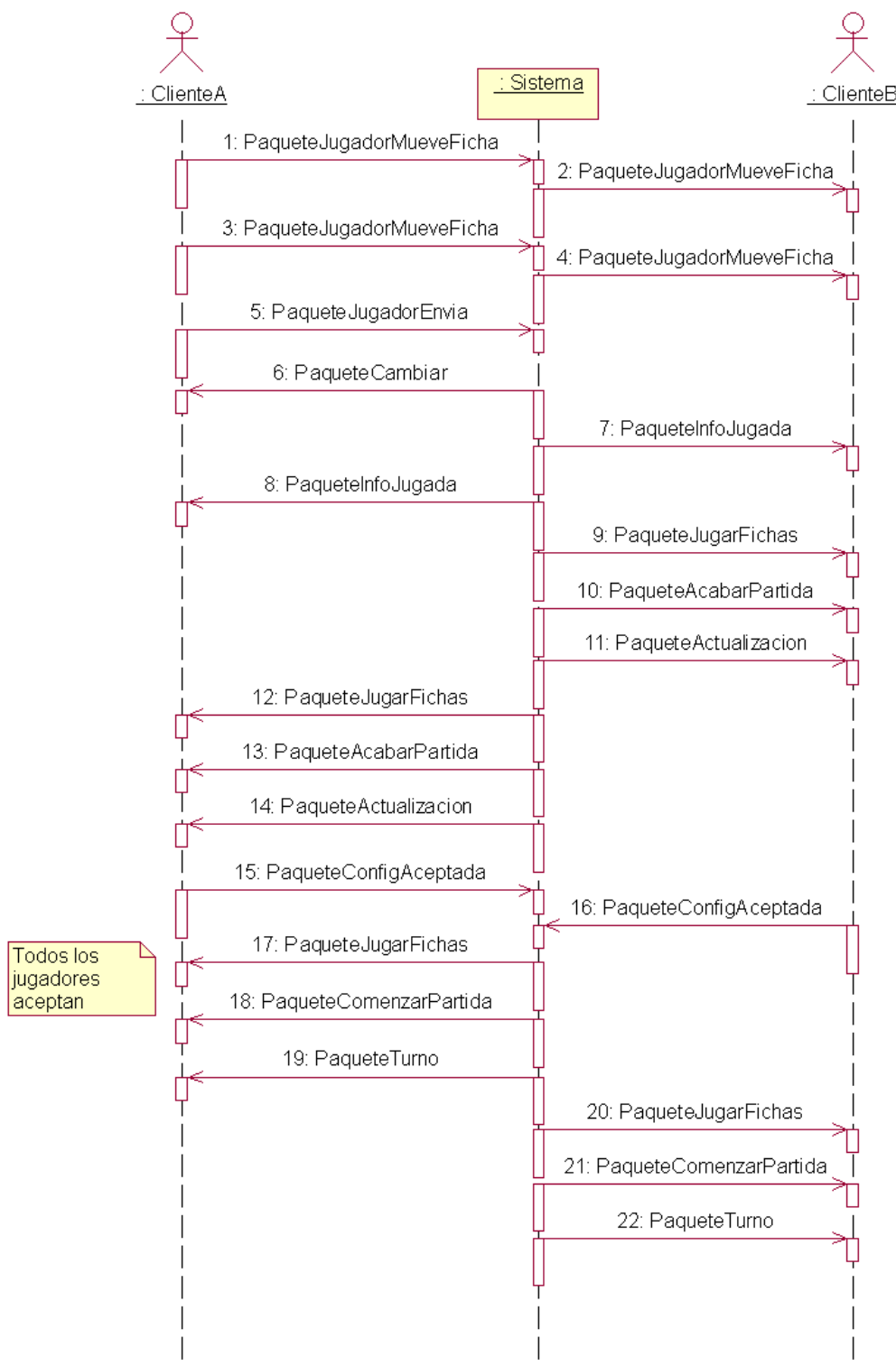


Ilustración 4.44 Diagrama de Secuencia Acabar Partida (Jugadores aceptan)

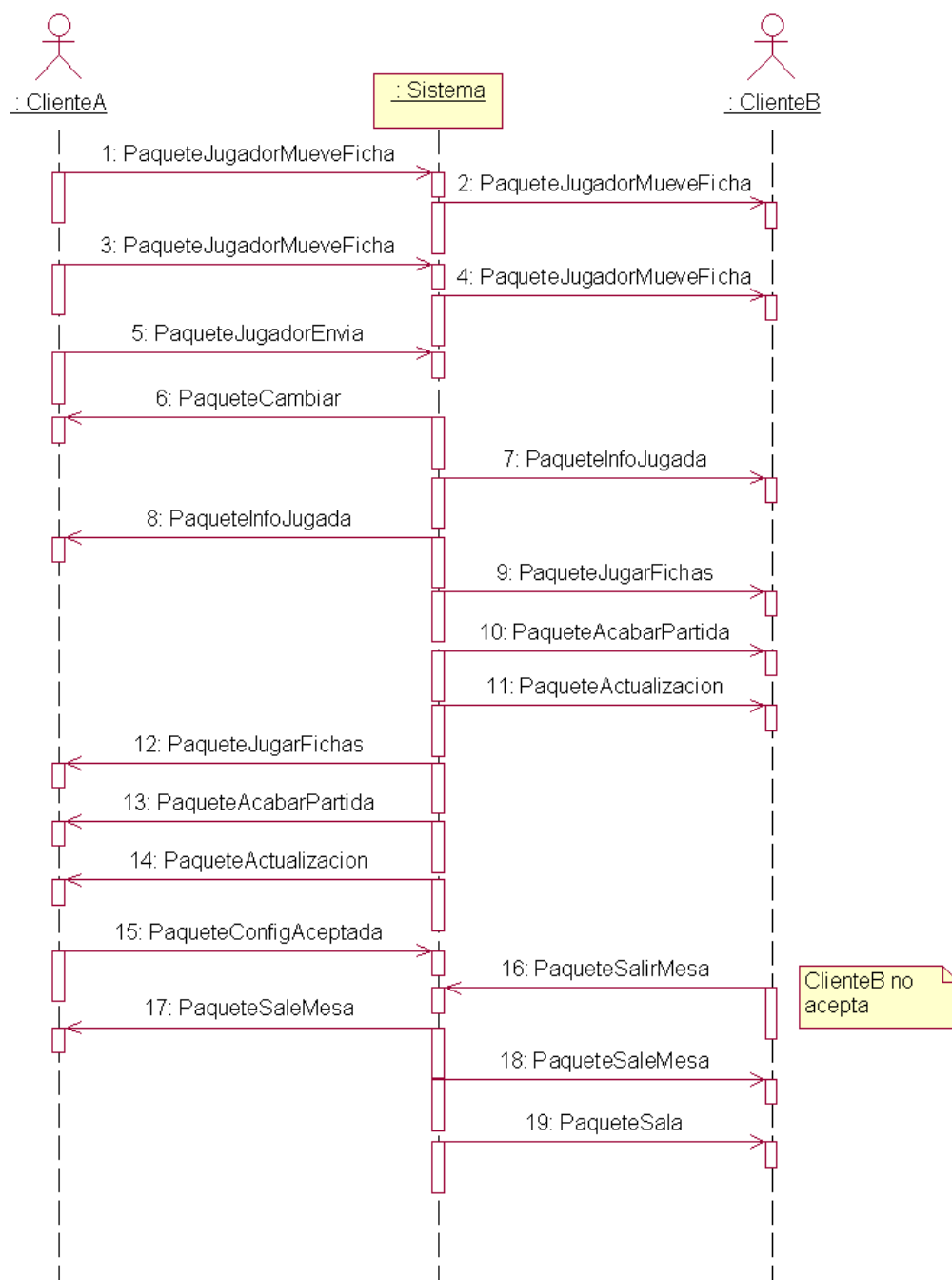


Ilustración 4.45 Diagrama de Secuencia Acabar Partida (Jugador rechaza)

4.2.6. Diagrama de Secuencia: Pausar Partida

Se distinguen dos situaciones diferentes: el Jefe de Mesa puede pausar directamente el juego, pero para el resto de jugadores, si desean pausar el juego, el Jefe de Mesa deberá aceptar si se pausa o no. Tanto el Jefe de Mesa como el resto de jugadores sólo disponen de tres peticiones de pausa (peticiones que, en el caso del Jefe de Mesa, serán concedidas en su totalidad).

En este caso, el Jefe de Mesa será ClienteA.

- Caso 1: El Jefe de Mesa Pausa

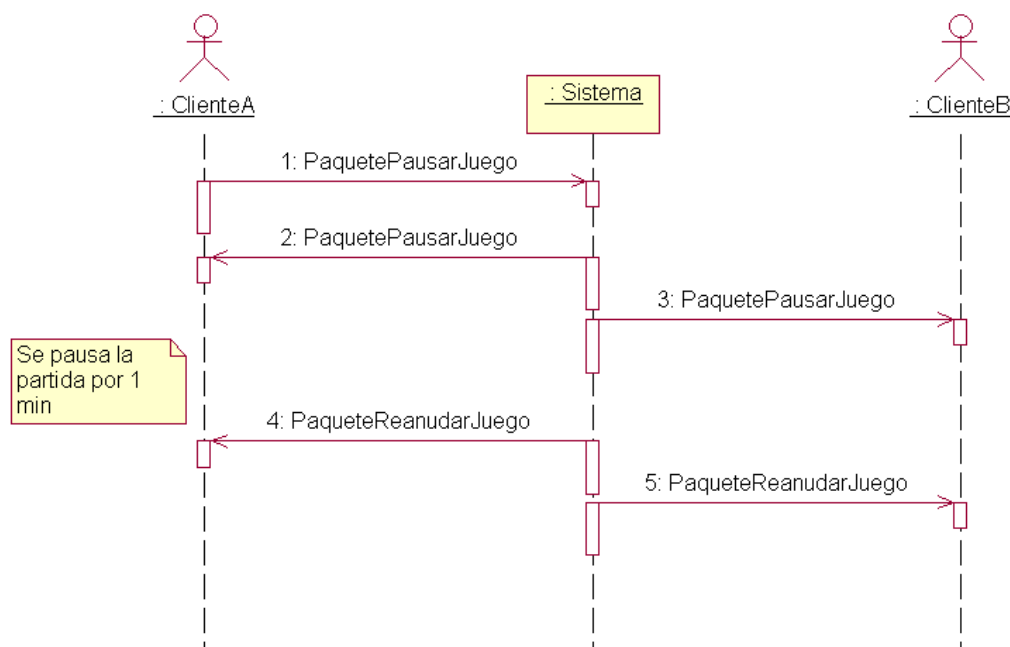


Ilustración 4.46 Diagrama de Secuencia Pausar Partida: Jefe de Mesa pausa I

En el caso de que se le agoten las pausas, se le mandará un mensaje de información:

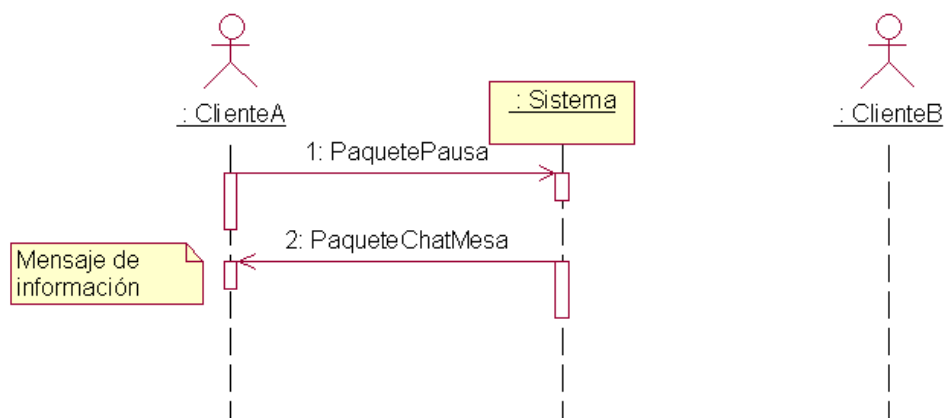


Ilustración 4.47 Diagrama de Secuencia Pausar Partida: Jefe de Mesa pausa II

· Caso 2: Un jugador Pausa

-Jefe de Mesa acepta

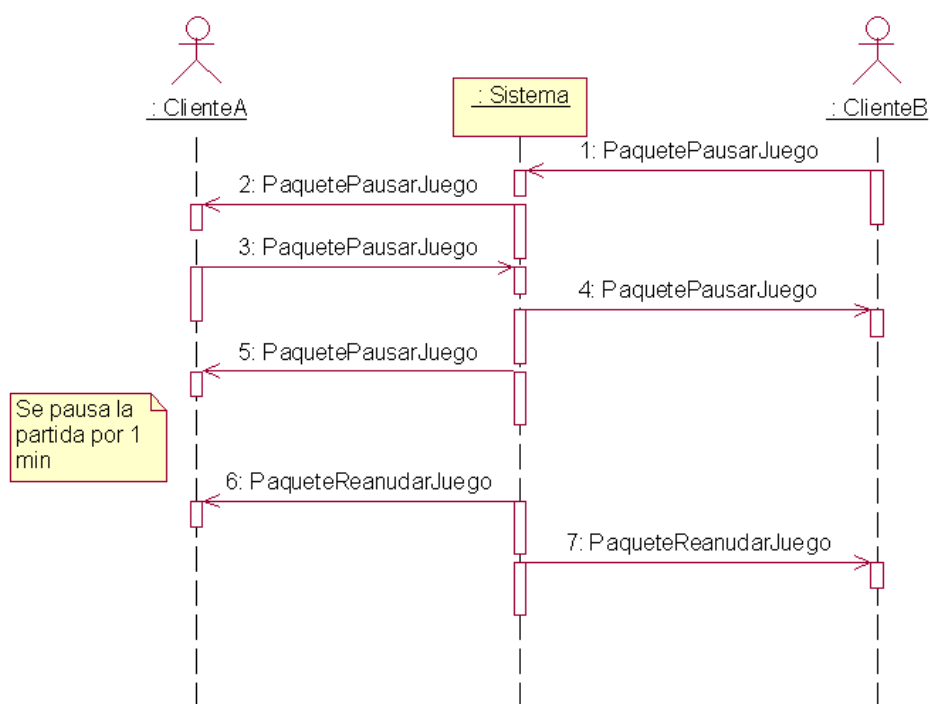


Ilustración 4.48 Diagrama de Secuencia Pausar Partida: Jugador pausa I

-Jefe de Mesa no acepta

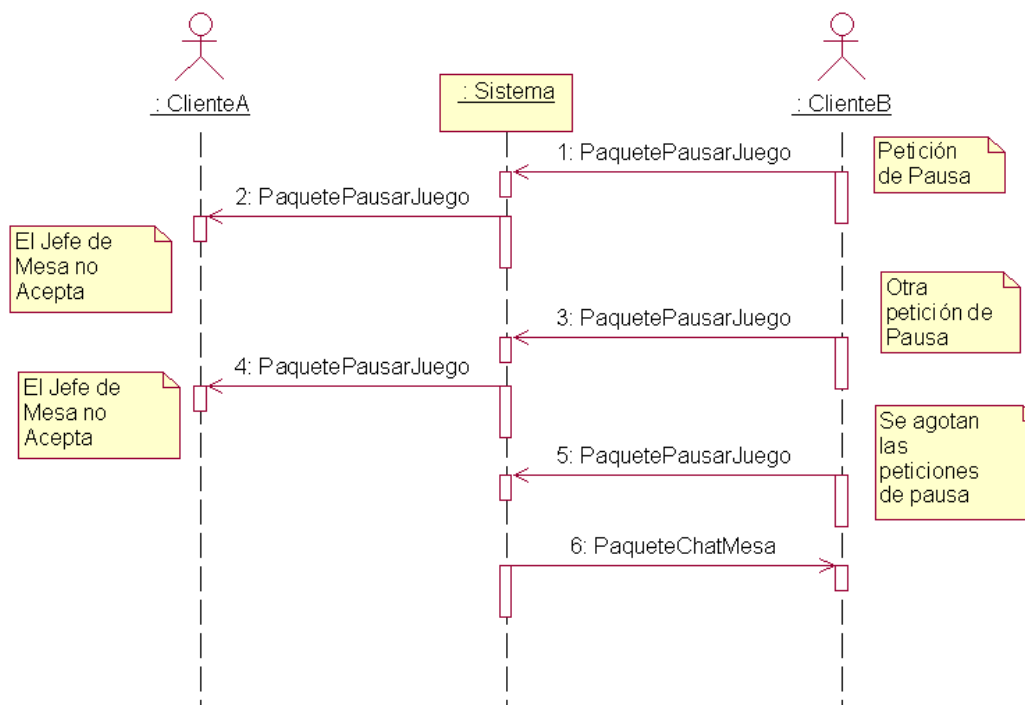


Ilustración 4.49 Diagrama de Secuencia Pausar Partida: Jugador pausa II

Cuando se agoten las peticiones de pausa del jugador, ya no se enviarán más paquetes al Jefe de Mesa para que valide la petición, y se le enviará un mensaje al jugador informándole de que no dispone de más pausas.

4.2.7. Diagrama de Secuencia: Abandonar

En el caso de que un jugador abandone la mesa, ya sea antes de que comience la partida o durante el transcurso de ella, la secuencia de paquetes es la siguiente:

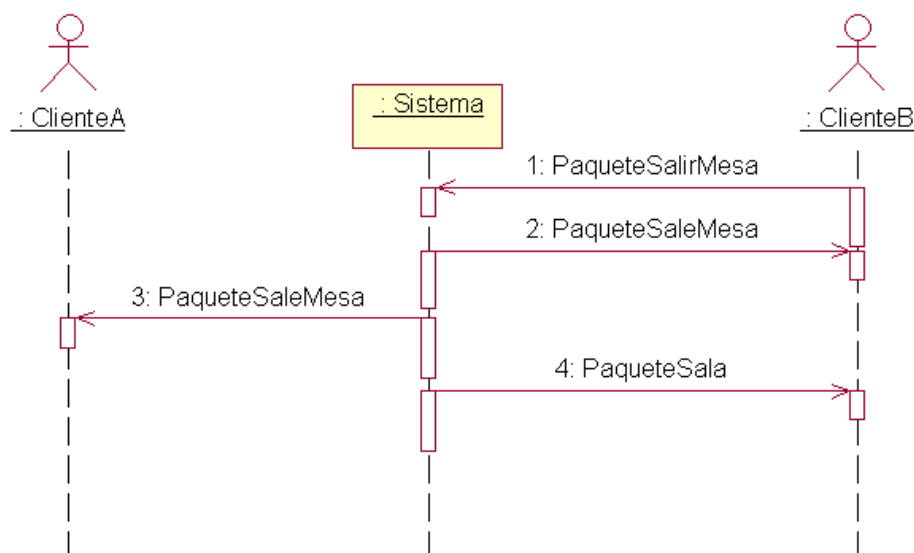


Ilustración 4.50 Diagrama de Secuencia Abandonar

Será en el servidor, en la parte de procesar el paqueteSalirMesa, donde se verá si había una partida en curso y actuar en consecuencia.

4.2.8. Diagrama de Secuencia: Salir del Sistema

Como se observa, cuando un jugador se desconecta, no se envía paquete alguno. Si se produce una desconexión, en el Servidor se interrumpen los hilos Enviador y Recibidor; cuando esto ocurra, se realizarán las acciones oportunas según el caso.

En el primer diagrama, se observa cómo se comporta la aplicación según los distintos casos:

- Si un jugador del Salón abandona el Sistema, sólo se informará a todos los jugadores del Salón.
- Si un jugador de una determinada Sala abandona el Sistema, sólo se informará a todos los jugadores de la Sala.

En el segundo diagrama, muestra el caso de dos jugadores de una mesa que han aceptado la configuración del juego (partida comenzada).

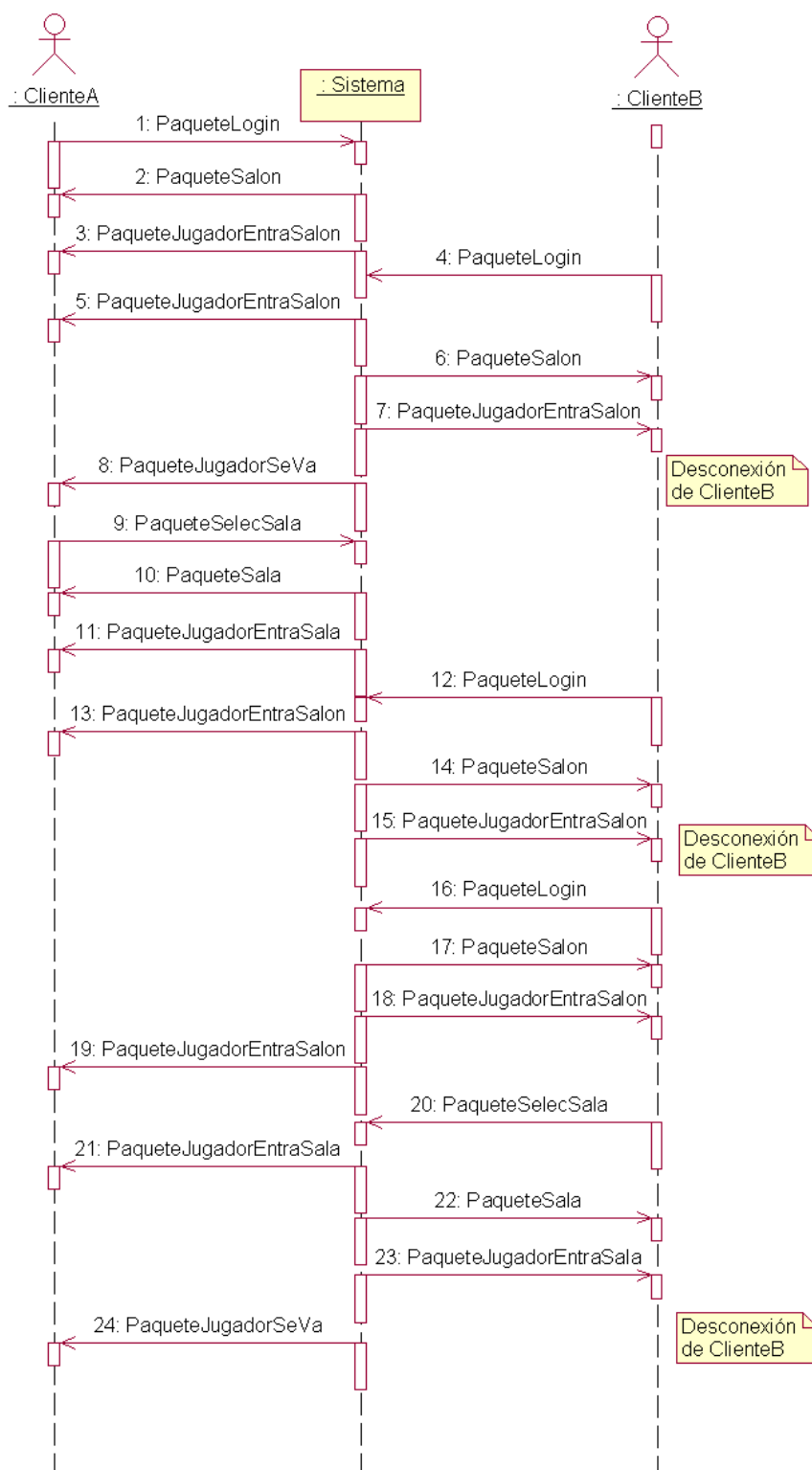


Ilustración 4.51 Diagrama de Secuencia Salir Sistema I

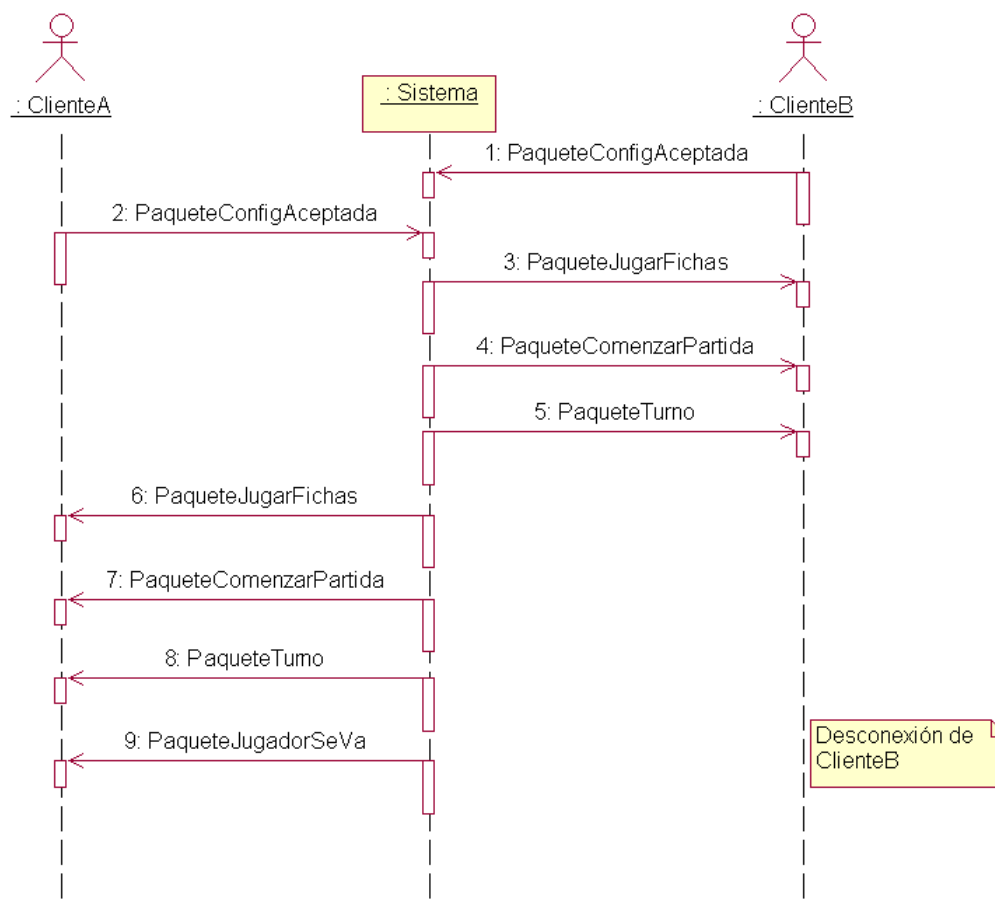


Ilustración 4.52 Diagrama de Secuencia Salir Sistema II

CAPÍTULO 5: IMPLEMENTACIÓN Y PRUEBAS

5.1. IMPLEMENTACIÓN

• Límite de Clientes

Se decidió controlar el número de usuarios que puedan utilizar simultáneamente la aplicación para obtener un funcionamiento óptimo del sistema.

Se cuenta con:

- Cuatro Salas de mesas para dos jugadores, con quince mesas.
- Tres Salas de mesas para tres jugadores, con diez mesas.
- Dos Salas de mesas para cuatro jugadores, con ocho mesas.

El número máximo de clientes permitido será pues de **274**.

• Sincronización

Para cada petición de conexión de un Cliente aceptada, el Servidor crea dos hilos; debido al gran número de hilos que comparten los mismos recursos, se ha puesto especial empeño en la sincronización de hilos para garantizar el buen funcionamiento de la aplicación.

La forma básica de conseguir la sincronización cuando varios hilos puedan modificar un recurso es mediante el uso de la clase Vector. La clase Vector es exactamente igual a la clase ArrayList pero además tiene sincronizados todos sus métodos (Thread Safe). Este comportamiento no incluye la sincronización sobre el recorrido de la lista, por lo que esto se hará de forma “manual” (mediante el bloque synchronized).

Las listas “conflictivas” (que se modifican a lo largo de la ejecución por varios hilos distintos) son las listas de jugadores del Salón, la Sala y la Mesa, así como la lista de clientes del sistema.

```
public class Servidor {  
    public final Vector<ClienteHandler> clientes;  
  
    public final Vector<JugadorInfo> jugadores; //jugadores del Salón
```

Ilustración 5.1 Declaración de las listas de clientes del sistema y jugadores del Salón

```
public class Sala {  
  
    public final Vector<JugadorInfo> listaJugadores;
```

Ilustración 5.2 Declaración de la lista de jugadores de la Sala

```
public class MesaInfo {  
    public final Vector<JugadorInfo> jugadores;
```

Ilustración 5.3 Declaración de la lista de jugadores de la mesa

• Control del Tiempo

Al tratarse de un juego por turnos con una duración máxima, era necesario desarrollar una forma de controlar el tiempo de los turnos. Además, como ya se ha visto, existe una opción para pausar la partida durante un minuto. Se optó por la utilización de la clase **Timer** y de **TimerTask**.

A continuación un ejemplo, en la clase **Partida** del **Servidor**:

```
public class Partida {  
  
    public int tiempo;  
    public int tiempoActual;  
  
    public Timer timer;  
    public MesaInfo mesaInfo;  
  
    public boolean pausada;  
    //Este flag indicará si se ha procedido a pausar la partida  
    public int tiempoPausa;  
    public Timer timerPausa;  
  
    public Partida(MesaInfo mesa) {  
        this.mesaInfo = mesa;  
  
        tiempo = mesaInfo.tiempo;  
  
        tiempoActual = tiempo;  
  
        timer = new Timer();  
  
        pausada = false;  
        tiempoPausa = 60;  
        timerPausa = new Timer();  
    }  
}
```

Ilustración 5.4 Implementación Clase Partida

La clase interna **RemindTask**, sobrescribe el método **run** de la clase **TimerTask**, para poder realizar una tarea de manera automática y repetitiva como se verá más adelante.

```
class RemindTask extends TimerTask{
    public void run(){
        if (!pausada) {
            //El tiempo correrá siempre que no esté pausada la partida
            if (tiempoActual > 0) {
                tiempoActual--;
            }
            else{
                //Agotado el tiempo, cambio de turno
                synchronized(mesaInfo.jugadores){
                    String anteriorJugador = mesaInfo.jugadores.get(mesaInfo.jugadorActual).nombre;
                    mesaInfo.mesa.cambiarTurno();
                    PaqueteTurno paqueteTurno = new PaqueteTurno();
                    paqueteTurno.nombreJugador = mesaInfo.jugadores.get(mesaInfo.jugadorActual).nombre;
                    paqueteTurno.mensaje = anteriorJugador + " pasa turno";

                    for(JugadorInfo j : mesaInfo.jugadores) {
                        j.cliente.enviador.enviar(paqueteTurno);
                    }
                }
            }
        }
    }
}
```

Ilustración 5.5 Clase interna RemindTask

El funcionamiento es el siguiente:

Cuando dé comienzo una partida, se llamará al método **iniciarTiempo()**: este método creará un nuevo timer, que cada 1000 milisegundos, es decir, cada segundo, ejecutará la tarea RemindTask. Cada vez que esto ocurra, se decrementará en una unidad **tiempoActual** (que comienza valiendo 30, 45 o 60, según el tiempo por turno elegido). Así, por ejemplo, al cabo de 30 ejecuciones, habrán pasado 30 segundos. Cuando **tiempoActual** valga 0, significará que el tiempo del turno se ha agotado y se producirá un cambio de turno.

En **cambiarTurno()**, entre otras cosas, se llamará al método **reiniciarTiempo()**, que restablece el valor de tiempoActual para volver a decrementarse cada segundo, simulándose así un nuevo turno completo.

Asimismo, cuando un jugador pulse Enviar, Cambiar o Pasar, también se cambiará el turno y se llamará a **reiniciarTiempo()**.

Cuando una partida finalice o se interrumpa (por el abandono de un jugador), se llamará al método **pararTiempo()**, que finalizará el timer, desechando la tarea que tiene programada.

```
public void iniciarTiempo() {
    timer = new Timer();
    pausada = false;
    timer.schedule(new RemindTask(), 0, 1*1000);
    LogFrame.log("*****Inicio tiempo partida");
}

public void reiniciarTiempo() {
    tiempoActual = tiempo;
    LogFrame.log("*****reinicio tiempo partida");
}

public void pararTiempo() {
    timer.cancel();
    LogFrame.log("*****parar tiempo partida");
}
```

Ilustración 5.6 Métodos para iniciar, reiniciar y parar tiempo partida

Para pausar la partida, se ha desarrollado una estrategia similar, con una tarea que se ejecuta cada segundo hasta 60 veces (un minuto de pausa).

```
public void pausar() {
    pausada = true;
    iniciarTiempoPausa();
}

public void reanudar() {
    pausada = false;
}
```

Ilustración 5.7 Métodos para activar y desactivar la pausa

Cada vez que se pause la partida, **pausada** tomará el valor de **true**. De esta forma, cada vez que se ejecute la tarea **RemindTask**, no se decrementará **tiempoActual**.

```
class RemindTaskPause extends TimerTask{
    public void run() {
        if (tiempoPausa > 0) {
            tiempoPausa--;
        } else {
            reanudar();
            pararTiempoPausa();
            PaqueteReanudarJuego paqueteReanudarJuego = new PaqueteReanudarJuego();
            paqueteReanudarJuego.jugadorTurno = mesaInfo.jugadores.get(mesaInfo.jugadorActual).nombre;
            synchronized(mesaInfo.jugadores){
                for(JugadorInfo j : mesaInfo.jugadores){
                    j.cliente.enviador.enviar(paqueteReanudarJuego);
                }
            }
        }
    }

    public void iniciarTiempoPausa(){
        reiniciarTiempoPausa();
        timerPausa = new Timer();
        timerPausa.schedule(new RemindTaskPause(),0, 1*1000);
    }

    public void reiniciarTiempoPausa(){
        tiempoPausa = 60;
    }

    public void pararTiempoPausa(){
        timerPausa.cancel();
    }
}
```

Ilustración 5.8 Clase Interna RemindTask y métodos para iniciar, parar y reiniciar el tiempo de pausa

Además, se llamará a **iniciarTiempoPausa()** donde se creará un timer al que se le programará una tarea cada segundo que vaya decrementado **tiempoPausa** (inicializado a 60 con **reiniciarTiempoPausa()**), hasta llegar a cero. Entonces, se reanudará la partida y se llamará a **pararTiempoPausa()**, que finalizará el timer de la pausa y restablecerá **pausada** a **false**. De este modo **tiempoActual** seguirá decrementándose en la ejecución de la tarea RemindTask.

Además de para la simulación de los turnos y las pausas, se han utilizado timers para la realización de **timeouts**. Un ejemplo: Cada vez que vaya a comenzar una partida nueva, los jugadores dispondrán de un minuto para aceptar o rechazar la configuración. Si pasado este tiempo un jugador no ha aceptado, saldrá automáticamente de la Mesa.

· Gráficos

Para el interfaz de usuario básico se ha usado Java Swing (tablas, etiquetas, botones, textField, textArea...). Para la utilización de tablas, se ha optado por utilizar el componente JTable y utilizar como modelo de la tabla implementaciones propias a partir de la clase AbstractTableModel.

Para el tablero, al tener una funcionalidad muy específica y compleja, se ha utilizado un API a más bajo nivel (AWT), para dibujar líneas, imágenes... y gestionar los eventos del ratón.

• Applet

Uno de los objetivos de este proyecto se trataba de la utilización de Applets de Java para que los usuarios no tuvieran la necesidad de descargarse la aplicación y pudieran jugar desde casi cualquier plataforma.

Primero se programó la parte del Cliente como una aplicación Java Swing normal y, cuando se comprobó que todo funcionaba bien, se convirtió en un applet para que se pudiera embeber en un navegador, con ligeras modificaciones:

- Se creó un JApplet que sirviera de contenedor para todos los frames principales (sobreescribiendo los métodos setVisible de los frames para que en lugar de mostrar su contenido en pantalla, fijen su contenido como el contenido del Applet).
- No utilizar el método setAlwaysOnTop para las ventanas emergentes.
- No utilizar nunca el método exit o similares.
- No conectar con direcciones distintas a la del servidor desde donde se descargó el Applet.

```
try{  
    socket = new Socket(MainApplet.applet.getCodeBase().getHost(), 8189);  
    desconectado = false;  
} catch (Exception ex) {  
    ex.printStackTrace();  
}
```

Ilustración 5.9 Apertura de un socket con el host con la URL del directorio que contiene el applet, y el puerto

- Utilizar el método getImage de los Applets para cargar las imágenes.
- No acceder a archivos que estén fuera del jar.

5.2. DESPLIEGUE

Los pasos que se han realizado para el despliegue del sistema son:

- 1- Preparación del hardware del servidor
- 2- Instalación del sistema operativo:
Sistema Operativo Windows 2003

- 3- Instalación de los drivers

- 4- Instalación el entorno XAMPP:

El en el panel de control de xampp se pondrá que se ejecutan como servicio Apache y MySQL.

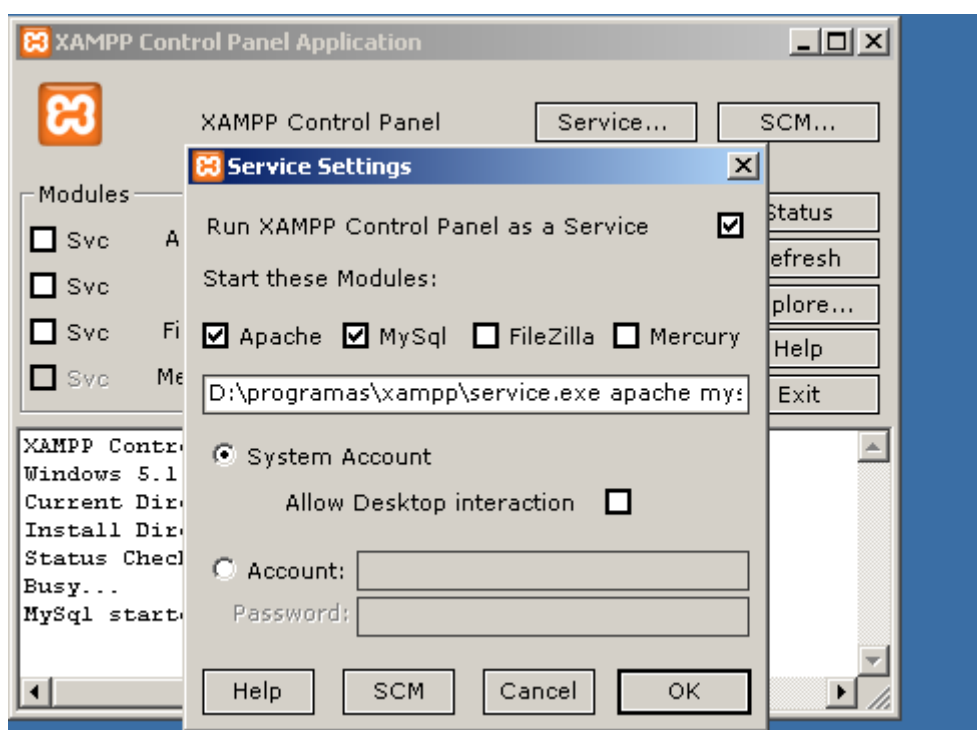


Ilustración 5.10 XAMPP: Ejecución de Apache y MySQL como servicio

Se añade contraseña a ambos para que no se puedan administrar remotamente si no es de forma autorizada.

```

1  <?php
2  /*
3   * This is needed for cookie based authentication to encrypt password in
4   * cookie
5   */
6  $cfg['blowfish_secret'] = 'xampp'; /* YOU SHOULD CHANGE THIS FOR A MORE SECURE COOKIE AUTH! */
7
8  /*
9   * Servers configuration
10  */
11  $i = 0;
12
13  /*
14   * First server
15  */
16  $i++;
17
18  /* Authentication type and info */
19  $cfg['Servers'][$i]['auth_type'] = 'http';
20  $cfg['Servers'][$i]['user'] = 'root';
21  $cfg['Servers'][$i]['password'] = 'root';
22  $cfg['Servers'][$i]['AllowNoPassword'] = true;
23
24  /* User for advanced features */
25  $cfg['Servers'][$i]['controluser'] = 'pma';
26  $cfg['Servers'][$i]['controlpass'] = 'pma';
27
28  /* Advanced phpMyAdmin features */
29

```

Ilustración 5.11 XAMPP: Añadir contraseña para restringir acceso

5- Despliegue de la página Web:

Se copia la página Web al directorio desde donde se sirven las páginas HTML por Apache.

Se crea la base de datos para la aplicación y se crea el usuario de ella para que lo usen los scripts PHP:

Servidor: localhost Base de datos: scrabble Tabla: jugador

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> login	varchar(12)	latin1_swedish_ci		No			[Iconos de acción]
<input type="checkbox"/> pass	varchar(30)	latin1_swedish_ci		Si	NULL		[Iconos de acción]
<input type="checkbox"/> fecha	date			Si	NULL		[Iconos de acción]
<input type="checkbox"/> puntuacion	int(11)			Si	0		[Iconos de acción]
<input type="checkbox"/> partidas	int(11)			Si	0		[Iconos de acción]
<input type="checkbox"/> ganadas	int(11)			Si	0		[Iconos de acción]
<input type="checkbox"/> perdidas	int(11)			Si	0		[Iconos de acción]
<input type="checkbox"/> abandonos	int(11)			Si	0		[Iconos de acción]
<input type="checkbox"/> elo	int(11)			Si	1000		[Iconos de acción]

Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Ilustración 5.12 Base de datos: Tabla Jugador

Vista global de usuarios

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [Mostrar todo]

	Usuario	Servidor	Contraseña	Privilegios globales ¹	Conceder	
<input type="checkbox"/>	cualquiera	%	--	USAGE	No	
<input type="checkbox"/>	cualquiera	localhost	No	USAGE	No	
<input type="checkbox"/>	pma	localhost	Sí	RELOAD, SHUTDOWN, PROCESS	No	
<input type="checkbox"/>	root	127.0.0.1	Sí	ALL PRIVILEGES	Sí	
<input type="checkbox"/>	root	localhost	Sí	ALL PRIVILEGES	Sí	
<input type="checkbox"/>	scrabble	localhost	Sí	USAGE	No	

⬅ Marcar todos/as / Desmarcar todos

Ilustración 5.13 Base de datos: Creación de un usuario

6- Ejecución del servidor de juego:

Se instala el JRE (Java Runtime Environment) y se copia el servidor Java a una carpeta del servidor. Se configura una tarea (Mediante el Task Scheduler) para que se ejecute siempre que se inicia el ordenador.

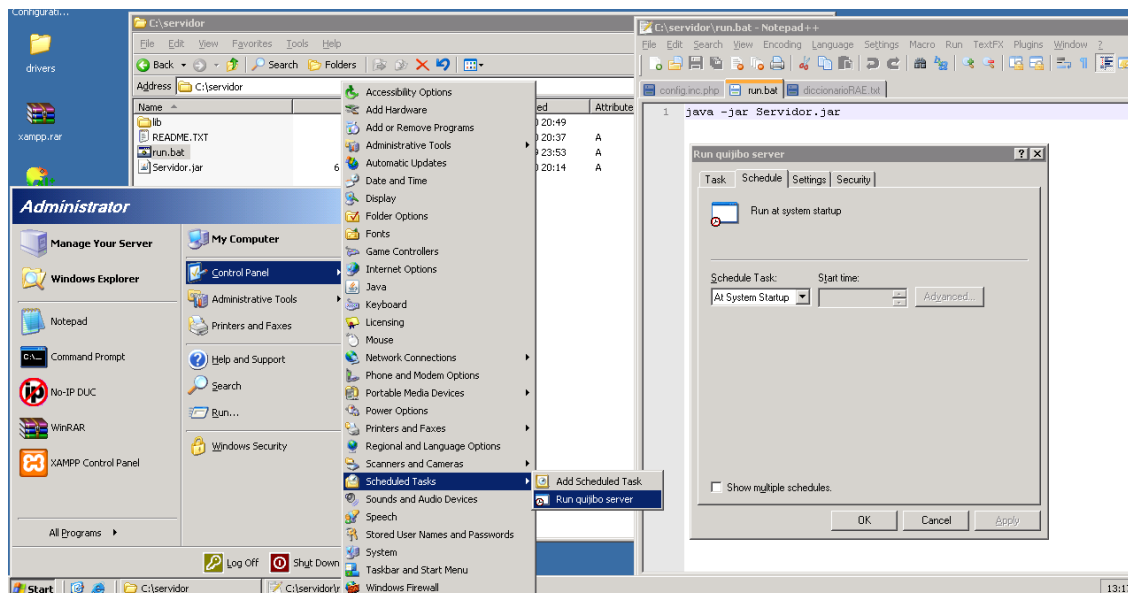


Ilustración 5.14 Ejecución del servidor de juego

7- Configuración de una cuenta de DNS dinámico:

Ya que la IP con la que cuenta el servidor es dinámica, se utiliza una cuenta gratuita de NO-IP para que provea de un nombre DNS que apunte a la IP. Además se instalará un programa proporcionado por no-IP que permitirá que el servidor comunique a no-IP de la IP si esta sufre cambios (de esta forma el DNS siempre apunta al servidor).

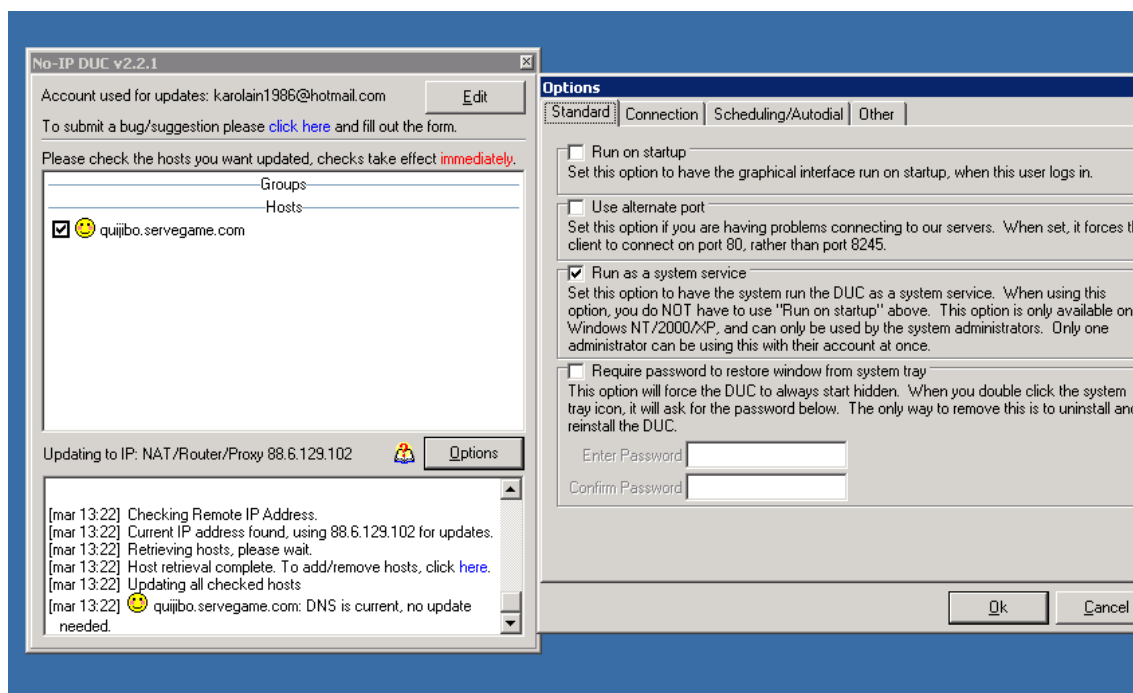


Ilustración 5.15 Programa que comunica de los cambios de la IP

8- Abrir los puertos del router:

Hace falta configurar los puertos a los que se permite acceder desde el exterior hacia el servidor, estos son:

Web

FTP

Escritorio remoto

Puerto al que se conectan los clientes (puerto del servidor de juego Java).

5.3. PRUEBAS

Después del despliegue, se llegó a la parte de las pruebas; en ellas, se solucionaron algunos pequeños fallos y problemas de la aplicación. Finalmente, se probó el sistema con un grupo grande de usuarios jugando simultáneamente con un resultado satisfactorio. Actualmente se puede probar su funcionamiento en:

<http://quijibo.servegame.com/>

CAPÍTULO 6: CONCLUSIONES Y LÍNEAS FUTURAS

A continuación, basándose en el desarrollo e implementación de la herramienta, se obtendrán las conclusiones y líneas futuras, que incluyen las posibles mejoras de la herramienta.

6.1. CONCLUSIONES

La realización de este proyecto ha consistido en el análisis, diseño e implementación de un sistema software. Este proyecto se puede dividir en dos partes: por un lado, el desarrollo de una página Web y por otro, la aplicación del juego, que se trata de la parte fundamental del proyecto y que se compone de dos partes a desarrollar: el servidor y el cliente.

Con este proyecto, se ha profundizado ampliamente en el conocimiento y uso del lenguaje de programación Java:

- La implementación de una aplicación cliente servidor, desarrollando ambas partes en su totalidad, utilizando hilos, sockets, streams...
- La creación de un protocolo propio para la comunicación del servidor con el cliente mediante el envío de paquetes.
- El manejo de un gran número de clases y de las relaciones entre ellas.
- La utilización de los Applets de Java para ejecutar la aplicación en un navegador Web.
- La interacción de Java con una base de datos.
- Una mayor profundización en el uso de interfaces gráficos de usuario tanto a alto nivel (Swing y componentes) como a bajo nivel (AWT y eventos).

Asimismo, se ha trabajado con los lenguajes HTML, PHP y JavaScript, de gran importancia debido al creciente desarrollo de las aplicaciones Web.

En resumen, el desarrollo del proyecto ha contribuido notablemente a ampliar la formación adquirida en la carrera, sobre todo en el lenguaje Java, uno de los más requeridos e importantes en el mundo del mercado de empleo.

6.2. LÍNEAS FUTURAS

Las mejoras que se podrían estudiar para aplicarse en un futuro a la aplicación del juego son las siguientes:

- La creación de Administradores que puedan expulsar a jugadores del sistema.
- Añadir la opción de que el Jefe de Mesa pueda expulsar a jugadores de la mesa.
- Que un jugador que esté en una mesa, pueda ver la lista de jugadores de la Sala y, si es Jefe de Mesa, invitarlos a jugar.
- La posibilidad de entrar a una mesa como espectador mientras se está disputando una partida; esta opción de permitir espectadores podrá ser habilitada o no por el Jefe de Mesa.
- Que en el caso de una desconexión accidental de un usuario (por ejemplo, se pierde la conexión a internet) mientras se estaba disputando una partida, se cree un timeout de unos minutos para que el usuario pueda volver a conectarse y, de este modo, se pueda reanudar la partida donde se había dejado. Una vez agotado el timeout, se consideraría que el jugador ha abandonado la partida y, si ésta se estaba disputando en Modo Competición, se le aplicaría la penalización pertinente.

Asimismo, en la parte de la Web, se podrían añadir más opciones:

- La creación de un foro en el que los usuarios registrados puedan solucionar dudas, comentar y dejar sugerencias; también la posibilidad de enviar mensajes privados a los demás usuarios registrados.
- Un sistema de registro de validación mediante correo que, además permitiría la recuperación de la contraseña en caso de olvido; además, periódicamente se mandaría un correo a los usuarios con novedades.
- Almacenar más datos de cada usuario como, por ejemplo, un avatar (imagen personalizable).

CAPÍTULO 7: BIBLIOGRAFÍA

- Libros:

- *Thinking in Java (4th Edition)* - Prentice Hall - Bruce Eckel
- *Core Java(TM), Volume I - Prentice Hall* - Cay S. Horstmann, Gary Cornell
- *Core Java, Vol. 2: Advanced Features* - Prentice Hall - Cay S. Horstmann, Gary Cornell
- *Java Game Programming for Dummies* - John Wiley & Sons Inc - Wayne Holder, Doug Bell
- *Killer Game Programming in Java - O'Reilly* - Andrew Davison
- *Java Network Programming - O'Reilly* - Elliotte Harold
- *HTML 4 For Dummies – For Dummies* - Ed Tittel, Mary Burmeister

Internet:

- <http://java.sun.com/j2se/1.5.0/docs/api/>
- <http://java.sun.com/applets/>
- <http://www.chuidiang.com/java/mysql/EjemploJava.php>
- <http://www.Webestilo.com/php/>
- http://juegos.orange.es/oniric2/ayuda/juegosmultijugador_index.html
- <http://es.wikipedia.org/wiki/Scrabble>
- <http://www.ajscrabble.org/>
- <http://es.wikipedia.org/wiki/Elo>
- <http://proton.ucting.udg.mx/tutorial/java/Cap2/holamapp.html>
- <http://www.cica.es/formacion/JavaTut/Cap2/escribir.html>
- <http://perso.wanadoo.es/evaristosaa/Applets.htm>
- <http://www.w3schools.com/html/>

- <http://apdaza-poo.blogspot.com/2007/09/programacin-de-un-timer-en-java.html>
- <http://www.javaranch.com/>
- http://www.laWebdelprogramador.com/news/mostrar_new.php?id=44&texto=Java&n1=409702&n2=1&n3=0&n4=0&n5=0&n6=0&n7=0&n8=0&n9=0&n0=0#
- <http://www.java2s.com/Code/JavaAPI/javax.swing.table/AbstractTableModel.htm>
- <http://java.sun.com/j2se/1.4.2/docs/api/java/lang/Class.html>
- <http://java.sun.com/javase/6/docs/api/java/util/concurrent/ArrayBlockingQueue.html>
- <http://www.programacion.com/java/tutorial/escribirapplets/29/>
- <http://www.iam.ubc.ca/guides/javatut99/uiswing/painting/loadingImages.html>

ANEXO: MANUAL DE USUARIO

En esta sección se describe el funcionamiento de la herramienta y de todos sus apartados, viendo en cada momento, todas las opciones disponibles para el usuario.

La Web

-Página Principal:

Una vez abierta la página Web en cualquier navegador de Internet, se mostrará la siguiente pantalla:

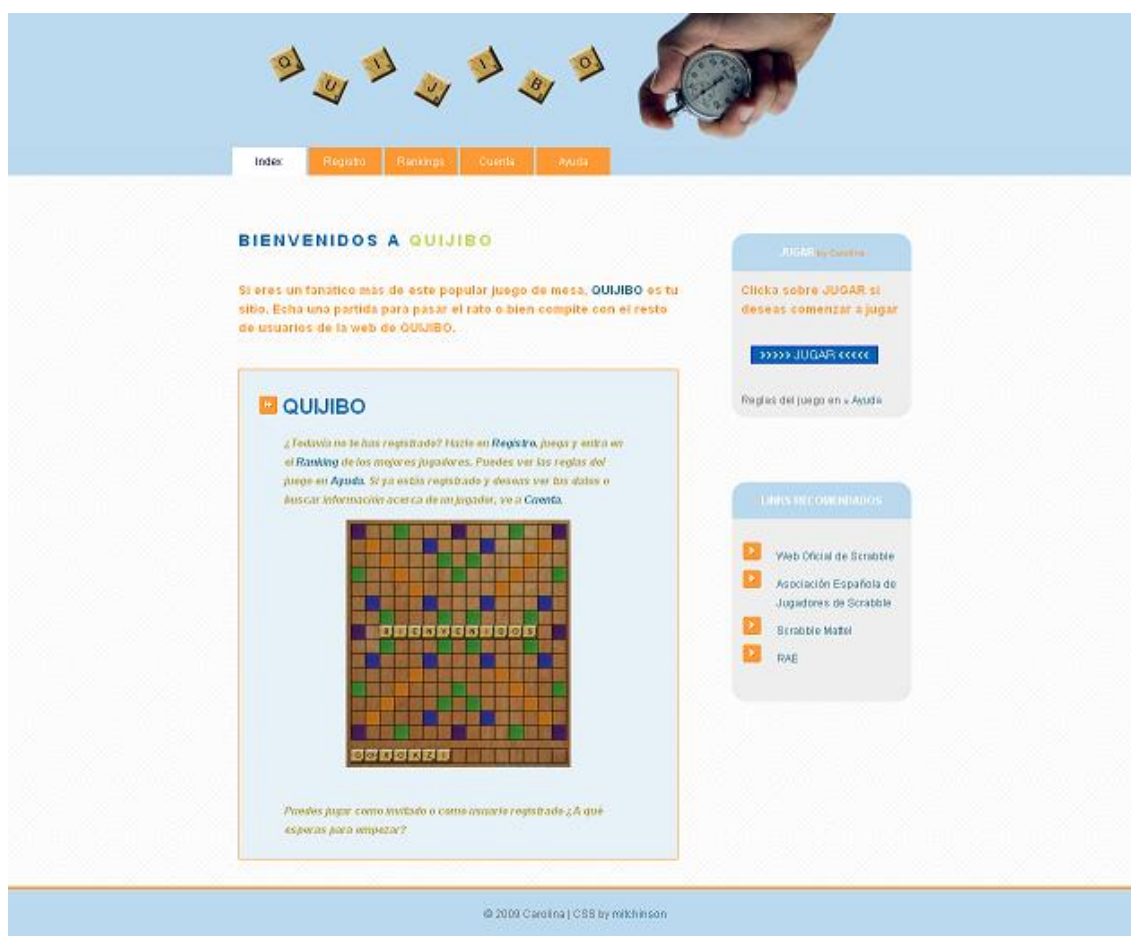


Ilustración A.1 Página Principal QUIJIBO

Pulsando el botón de la ilustración A.2, se ejecutará la aplicación:

>>>> JUGAR <<<<

Ilustración A.2 Botón JUGAR

-Registro:

Si el usuario desea registrarse, sólo tendrá que rellenar un breve formulario en esta sección.

REGISTRO

Registro

Datos

Por favor, introduce tus datos

nombre/nick:

contraseña:

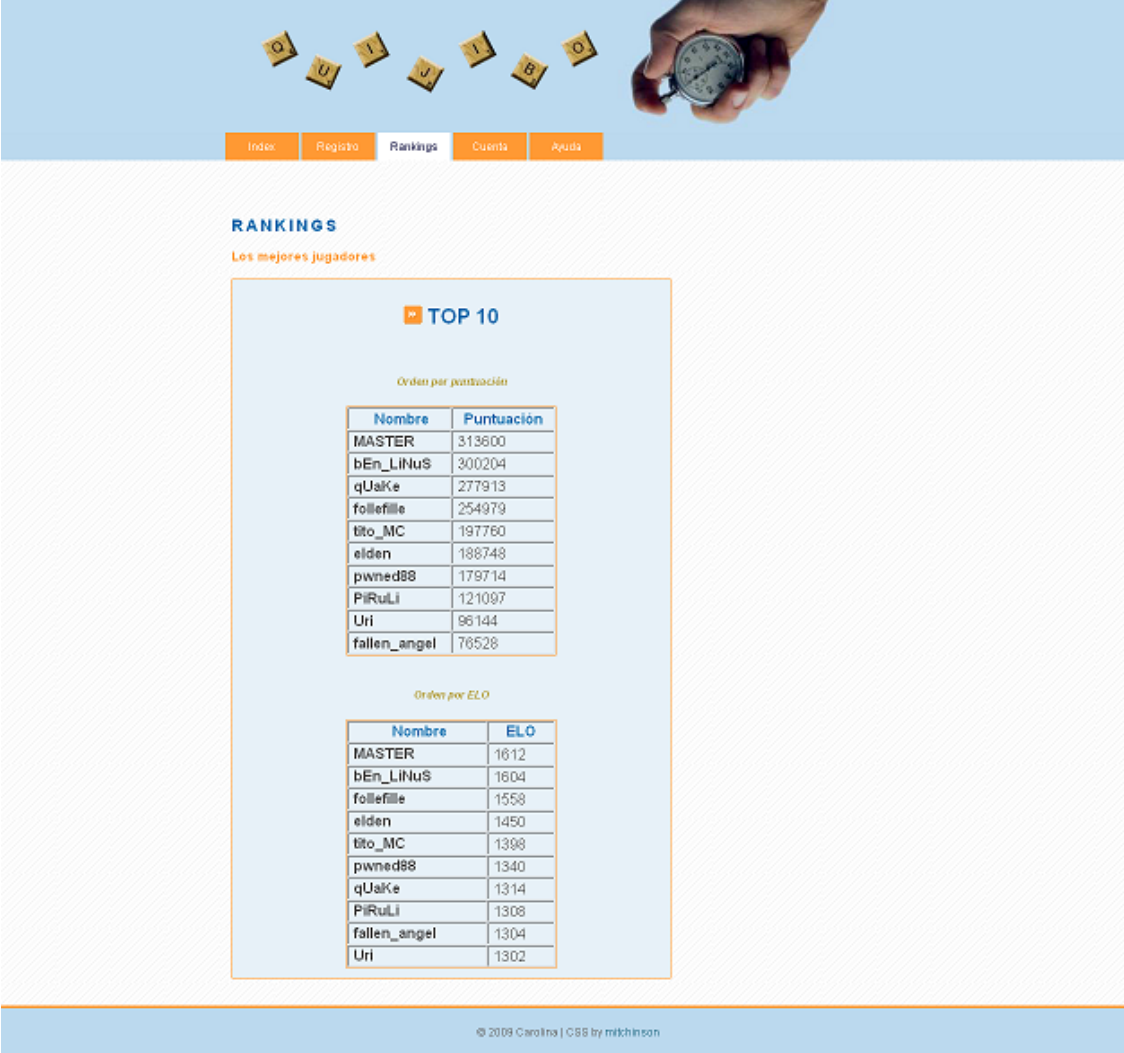
repetir contraseña:

Enviar

Ilustración A.3 Formulario de Registro

-Rankings:

En esta sección se mostrarán 2 lista con los nombres de los mejores jugadores registrados de QUIJIBO, atendiendo al número de puntos acumulados y a su ELO.



RANKINGS
Los mejores jugadores

TOP 10

Orden por puntuación

Nombre	Puntuación
MASTER	313600
bEn_LINuS	300204
qUaKe	277913
follefilie	254979
tito_MC	197760
elden	188748
pwned88	179714
PiRuLi	121097
Uri	96144
fallen_angel	76528

Orden por ELO

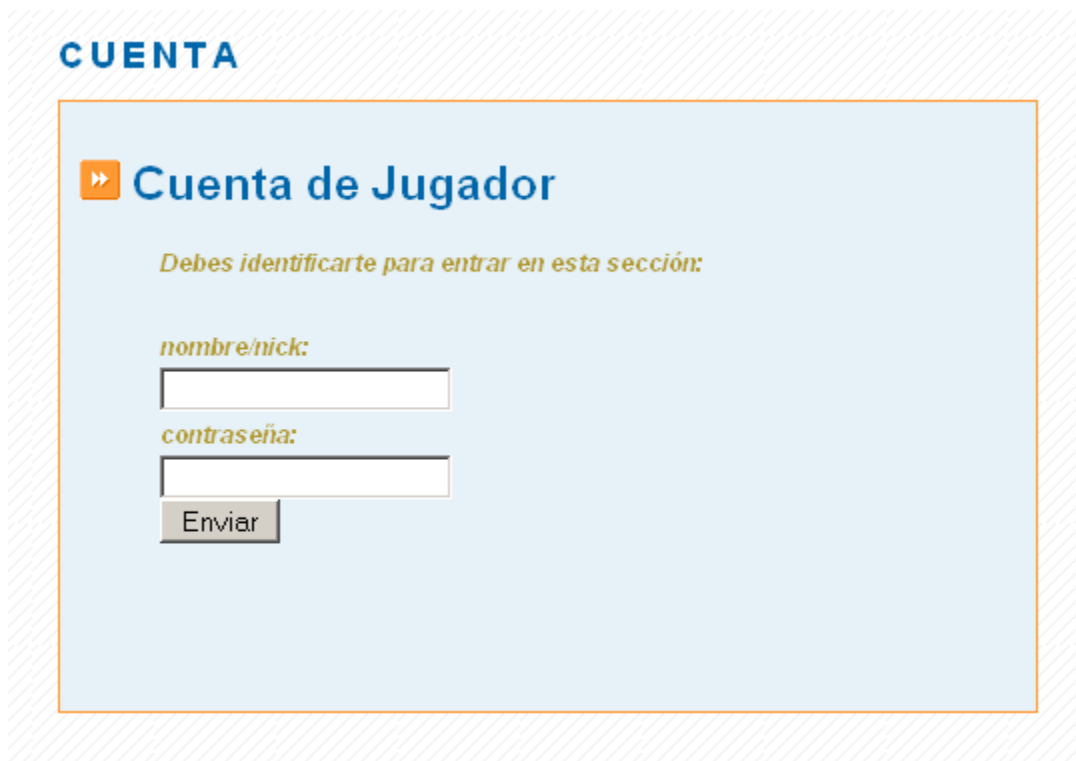
Nombre	ELO
MASTER	1612
bEn_LINuS	1604
follefilie	1558
elden	1450
tito_MC	1398
pwned88	1340
qUaKe	1314
PiRuLi	1308
fallen_angel	1304
Uri	1302

© 2009 Carolina | CSS by mikhlinson

Ilustración A.4 Rankings de jugadores

-Cuenta

Esta sección sólo será visible para los usuarios registrados, por lo que será necesario loguearse para entrar en ella.



The screenshot shows a web interface for a 'Cuenta de Jugador' (Player Account). At the top left, the word 'CUENTA' is written in blue. Below it, there is a blue header with an orange double arrow icon and the text 'Cuenta de Jugador'. Underneath the header, a message in orange italicized text says 'Debes identificarte para entrar en esta sección:'. There are two input fields: the first is labeled 'nombre/nick:' and the second is labeled 'contraseña:'. Below the password field is a grey button with the text 'Enviar'.

Ilustración A.5 Acceso a la Cuenta de Jugador

Una vez hecho esto, el usuario identificado podrá ver la información de su cuenta, así como realizar búsquedas de los demás usuarios registrados.

CUENTA

» Cuenta de Jugador

Tus Datos:

Nombre: karo

Fecha de registro: 2009-12-13

Puntuación general: 0

ELO: 1000

Partidas jugadas 0

Partidas ganadas 0

Partidas perdidas 0

Partidas abandonadas 0

Búsqueda de Jugador

Introduce su nick

OK

Ilustración A.6 Cuenta de Jugador



Ilustración A.7 Búsqueda de Jugador por nick

En cualquier momento, el usuario podrá desconectarse pulsando Logout en la esquina superior derecha de la página:



Ilustración A.8 Logout

La aplicación

A continuación se detallará una Ayuda sobre las reglas del juego y el funcionamiento de la aplicación.

1. Reglas del Juego

1.1. Disposiciones generales

Pueden jugar entre 2 y 4 jugadores, sobre un tablero de 15x15 casillas, en las que cada jugador coloca sus fichas. Cada jugador recibe un número específico de fichas (o letras). Las letras se encuentran numeradas con su respectivo valor, y por lo tanto, cada jugador obtiene por cada palabra formada un puntaje que depende tanto del valor de las letras empleadas como de la posición de dichas letras dentro del tablero.

En total hay 100 fichas, 98 marcadas con letras y dos en blanco (sin puntos, pero se pueden usar para reemplazar letras: se trata de fichas comodín). Según su frecuencia de aparición, las letras tienen más o menos puntos, siempre las de mayor frecuencia valen menos. El tablero tiene también casillas especiales, que multiplican el número de puntos concedidos: las casillas moradas son de "triple palabra", las naranjas "doble palabra", azul oscuro "triple letra" y verde "doble letra".

El juego consiste en formar palabras de dos o más letras y, colocarlas en el tablero, horizontal o verticalmente, de manera que puedan ser leídas de izquierda a derecha o de arriba hacia abajo. Después de la jugada inicial (que debe ocupar la casilla central), cada palabra sobre el tablero debe empalmarse con alguna de las incorporadas.

1.2. Desarrollo de la Partida

1.2.1. Reparto

Al comenzar la partida, se reparte a cada jugador siete (7) fichas del saco. Luego, el jugador que comienza combina dos o más de sus letras para formar una palabra, y las coloca en el tablero horizontal o verticalmente. El jugador debe poner una de las letras que forman su palabra en la casilla central.

1.2.2. Formación Palabras

Todas las fichas jugadas en un turno deben colocarse en una línea horizontal o vertical y tienen que formar una palabra completa. Si esas fichas están en contacto con letras de filas o columnas contiguas también con ellas deben formar palabras completas. El jugador obtiene puntos por todas las nuevas palabras que forme durante su turno.

1.2.3. Colocación Palabras

Las palabras pueden formarse de las siguientes maneras:

- Añadiendo una o más fichas a una palabra que está colocada sobre el tablero.
- Colocando una palabra que se cruce con otra en el tablero. La nueva palabra debe utilizar una de las letras de la palabra que ya se encuentra en el tablero.
- Colocando una palabra completa paralelamente a otra que ya se encuentra en el tablero, de modo que las fichas contiguas también forman palabras completas.

1.2.4. Comodines

Las dos fichas en blanco o comodines pueden utilizarse en sustitución de cualquier letra.

- El valor de los comodines es de cero puntos para calcular el valor de la(s) palabra(s) formada(s).
- Está permitido usar los dos comodines en una misma palabra. El premio de 50 puntos otorgado por utilizar las siete fichas del atril se contabiliza incluso utilizando uno o los dos comodines.
- Los comodines pueden reemplazar una letra ya utilizada, aunque exista solo una de ellas en el juego.

1.2.5. Dígrafos

Los dígrafos RR, CH, LL aparecen en una sola ficha. Por tanto, no podrán utilizarse dos eres, ni dos eles, ni la ce y la hache para formar una doble letra.

1.3. Recuento de Puntos

1.3.1. Valor numérico de las letras

El valor de cada letra está indicado con una cifra en la parte inferior de la ficha. La puntuación total obtenida en cada turno se calcula sumando el valor de las letras utilizadas en las palabras que se hayan formado en ese turno, más los puntos obtenidos por haber colocado una o más fichas en casillas con premio.

1.3.2. Casillas con premio para letras

Al colocar una ficha en una casilla verde (doble tanto de letra) se duplica el valor de dicha letra y al colocarla en una azul marino (triple tanto de letra) se triplica su valor.

1.3.3. Casillas con premio para palabras

Al colocar una palabra usando una casilla naranja (doble tanto de palabra) se duplica el valor de dicha palabra y al colocarla usando una casilla morada (triple tanto de palabra) se triplica su valor.

Al hacer el recuento de puntos de una palabra se deben sumar primero los premios de las letras y, luego, duplicar o triplicar el valor de la palabra, según sea el caso.

1.3.4. Palabras con valor múltiplo

Si se forma una palabra que ocupa dos casillas de "doble tanto de palabra", primero se duplica el valor de la palabra y luego se reduplica (es decir, se multiplica por 4 la suma de puntos de sus letras) y si son de "triple tanto de palabra", se triplica y luego se vuelve a triplicar (se multiplica por 9).

1.3.5. Casilla central

La casilla central es de color naranja, por lo que duplica la puntuación de la primera palabra formada en el tablero.

1.3.6. Unicidad de premios

Los premios para las letras y palabras aplican solamente la primera vez que se ocupa la casilla de premio. Posteriormente, las fichas que ocupan esas casillas cuentan simplemente por su puntaje real.

1.3.7. Palabras simultáneas

Cuando en un mismo turno el jugador forma dos o más palabras nuevas, se suma la puntuación de todas ellas. La(s) letra(s) comun(es) entre ellas se cuenta(n) en cada palabra, incluido el premio, si lo tiene(n).

1.3.8. Premio por "palabrón"

Cuando un jugador coloca las 7 fichas de su atril en un solo turno, ha logrado lo que se llama un palabrón y, por ello, obtiene 50 puntos extra como premio. Estos puntos se suman a la puntuación que haya obtenido en ese turno, después de duplicar o triplicar la puntuación, cuando corresponda.

1.4. Despliegue de Fichas en el Tablero

1.4.1. Casilla Central

Al inicio de la partida se debe ocupar la casilla central.

1.4.2. Comienzo de la jugada

El sentido de colocación de una palabra está dado por:

- La primera ficha colocada en el tablero define la fila o columna en que se colocará la palabra.

- La segunda ficha colocada define si la palabra ocupará una fila o una columna, la cual será de uso obligatorio para la jugada completa.
- También es posible colocar varias fichas simultáneamente, siempre que se cumpla con los puntos anteriores

1.4.3. Ficha colocada

Cuando el jugador en turno termina de colocar en el tablero las fichas de una palabra, puede devolverlas al reparto y colocar otras en su lugar, hasta que pulse el botón enviar o se le pase el turno; en este último caso, las fichas se devolverán al reparto y el jugador no puntuará.

1.5. Cambio de Fichas

1.5.1. Cambio de Fichas

Un jugador puede utilizar su turno para cambiar algunas fichas de su atril, o todas si así lo desea. Para ello, seleccionará la/s ficha/s que desea cambiar y pulsará el botón cambiar. Inmediatamente se le repartirán nuevas fichas. El jugador que ha cambiado fichas debe esperar al siguiente turno para jugar.

1.5.2. Turno cedido

Está permitido pasar en cualquier momento del juego, sin cambiar fichas, mediante el uso del botón Pasar.

1.6. Control del Tiempo

1.6.1. Duración de la Partida

La partida terminará dependiendo de cuál de los tres modos de juego se ha elegido en la configuración de la mesa:

Si se ha elegido el modo 150 puntos o 250 puntos, la partida terminará cuando uno de los jugadores alcance dicha puntuación.

Si se ha elegido el modo acabar fichas, la partida terminará cuando se acaben las fichas del saco.

1.6.2. Duración de un turno

La duración máxima del turno dependerá de cuál de los tres modos de tiempo se ha elegido en la configuración de la mesa: 1 minuto, 45 segundos o 30 segundos.

2. EMPEZAR A JUGAR

2.1. Sobre Jugadores Invitados y Registrados

En QUIJIBO, es posible jugar como jugador invitado o jugador registrado.

1-Accediendo como jugador invitado, se podrá jugar directamente sin detenerse en pasos previos. Sin embargo, no se guardarán las puntuaciones ni se calculará la experiencia del jugador, ni por tanto aparecerá en los rankings.

2- Registrándose en la Web, el usuario tendrá que rellenar un breve formulario de registro. A cambio, mantendrá su propio nick, así como un acumulado de puntos y estadísticas de los juegos.

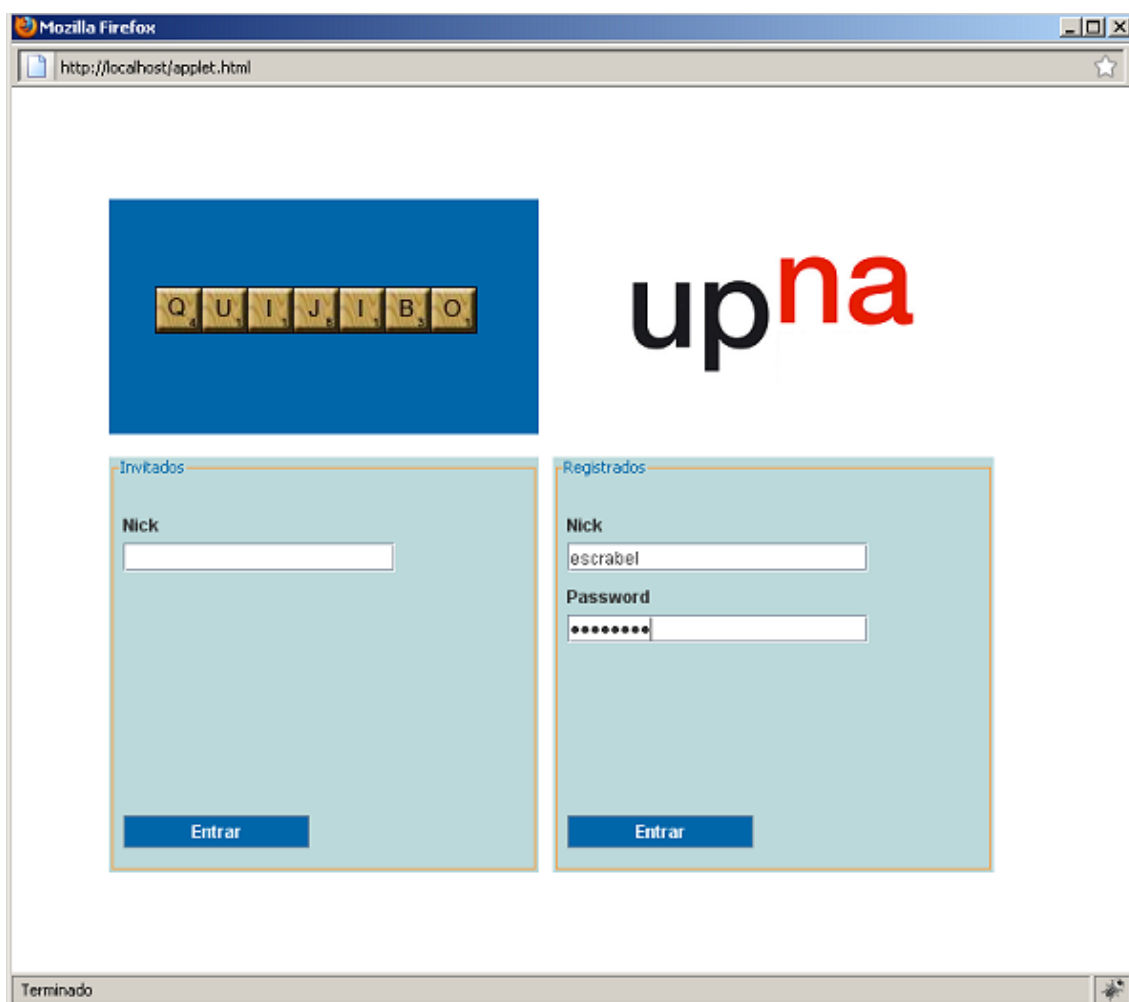


Ilustración A.9 Pantalla de Login

2.2. Salón Principal

Después de loguearse como invitado o como usuario registrado, el jugador entrará al Salón Principal.

En la parte superior derecha, aparece un listado de los jugadores que se encuentran en el Salón Principal, mostrándose el nick, la puntuación global y el ELO de cada jugador. Los jugadores registrados aparecerán de color azul, y los invitados de color blanco. Además, como se ha comentado, los invitados no tienen historial de juego. Debajo se encuentra el chat del Salón, con el que el jugador podrá comunicarse con el resto de jugadores que se encuentren en el Salón Principal.

En la parte izquierda aparecerá una tabla con el listado de Salas de Juego. Las Salas 0 a la 3 están diseñadas para partidas para dos jugadores; las Salas 4, 5 y 6, para tres jugadores; y finalmente, las salas 7 y 8, para cuatro jugadores.

Al lado de cada Sala aparece el aforo máximo y el número de jugadores que hay actualmente en cada una de ellas.

Para entrar en una sala, el jugador sólo tiene que seleccionarla y pulsar el botón Entrar.

The screenshot shows the Mozilla Firefox browser window with the address bar displaying 'http://localhost/applet.html'. The main content area is divided into several sections:

- Header:** A blue banner with the text 'QUIJIBO' in white letters.
- Game Rooms Table:** A table with three columns: 'Nombre', 'J.MAX', and 'Jugadores'. It lists rooms 0 through 8.
- Player Statistics Table:** A table with three columns: 'Nombre', 'P.Global', and 'ELO'. It lists players like 'java', 'hola', 'invit', 'ayayaya', 'quake88', 'mr_t', 'escrabel', 'bcn90', 'wewewe', and 'goku_jar'.
- Chat Log:** A text area showing a list of messages, including 'escrabel ha entrado al Salón', 'escrabel>hola', 'bcn90 ha entrado al Salón', 'bcn90>wenass', 'wewewe ha entrado al Salón', 'invit>alguien pa 1 d 4?', 'quake88>2 jugadores a 1 minuto', 'wewewe>tuuu javi', 'bcn90>sperate k ahora se mete ste', 'goku_jar ha entrado al Salón', 'goku_jar>ya', 'goku_jar>vamos pues?', and 'bcn90>si'.
- Room Descriptions:** A list of descriptions for different room types: 'Salas de 2 jugadores: 0,1,2,3', 'Salas de 3 jugadores: 4,5,6', and 'Salas de 4 jugadores: 7,8'.
- Buttons:** A blue button labeled 'Entrar' and a red button.
- Status Bar:** A bar at the bottom of the browser window showing 'Terminado'.

Ilustración A.10 El Salón Principal

2.3. Sala

Una vez seleccionada una sala, si el aforo máximo no estaba cubierto, el jugador entrará en ella.

Como en el Salón Principal, en la parte superior derecha aparece un listado de los jugadores que se encuentran únicamente en la misma Sala que el jugador. Debajo se encuentra el chat de la Sala

En la parte izquierda, aparece un listado de las mesas disponibles, mostrándose también el número de jugadores que se necesitan para una partida y el número de jugadores que hay actualmente en la mesa.

Si se selecciona una mesa, en la parte de abajo se mostrará el nick del/los jugador/es que se encuentra/n en la mesa.

Para entrar en una mesa, sólo hay que seleccionarla y pulsar el botón entrar.

Si se desea volver al Salón Principal, se pulsará el botón Volver.



Ilustración A.11 La Sala

2.4. Mesa

Ya se ha comentado que, dependiendo de la Sala, las mesas pueden ser de dos, tres o cuatro jugadores. Es decir, la partida no comenzará hasta que no se haya completado el número de jugadores necesarios.

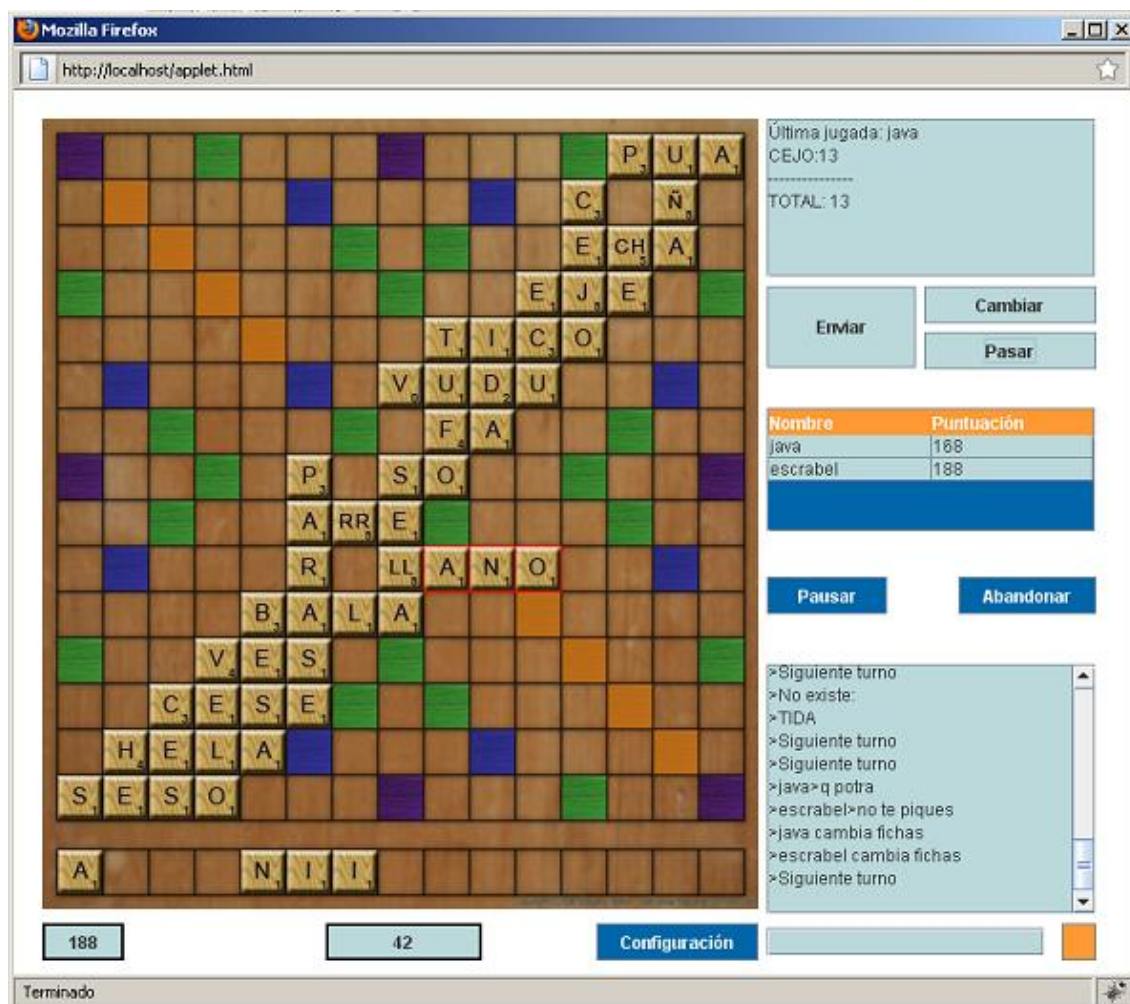


Ilustración A.12 La Mesa

2.4.1. El Jefe de Mesa

El primer jugador que entra en la mesa será el Jefe de Mesa. Tendrá ciertos privilegios, como la posibilidad de elegir la configuración del juego (sólo antes de comenzar la partida) o pausar la partida sin necesitar confirmación del resto de jugadores. También será suyo el primer turno en la primera partida.

Cuando un jefe de Mesa abandona la mesa, el nuevo Jefe de Mesa será el segundo jugador que entró a la mesa y, así, sucesivamente.

El resto de jugadores no podrán elegir la configuración del juego, deberán aceptar (o no) la elegida por el jefe de Mesa antes del inicio de la partida. Una vez comenzada, el jefe de Mesa no podrá cambiar la configuración.

2.4.2. El Tablero

Como se observa, la parte principal de la pantalla de la Mesa es el tablero. Debajo está el reparto del jugador con las siete fichas que le corresponden. Para jugar sus fichas, el jugador sólo tiene que llevarlas una por una de su reparto al tablero cuando sea su turno. Si aún no se ha pulsado el botón Enviar, puede devolver sus fichas al reparto.

2.4.3. Panel de Información

En la esquina superior derecha, aparece un panel de información, que mostrará de qué jugador fue la última jugada correcta, la/s palabra/s que formó y su correspondiente puntuación, así como la puntuación total obtenida.

2.4.4. Enviar, Cambiar y Pasar

Debajo del panel de información, están los tres botones principales de Juego: Enviar, Cambiar y Pasar.

Enviar: Una vez el jugador esté seguro de la/s palabra/s que ha formado sobre el tablero, deberá pulsar este botón para que la jugada sea válida. Si no lo hace, cuando se acabe su tiempo, las fichas se le devolverán al reparto y no puntuará.

Cambiar: Si el jugador desea cambiar alguna ficha del reparto, o la totalidad de ellas, sólo tiene que seleccionar cuáles y posteriormente pulsar el botón Cambiar. Las fichas serán devueltas al saco y se le asignarán nuevas fichas al jugador (cabe la posibilidad de que le vuelvan a tocar las mismas fichas; cuantas menos fichas haya en el saco, mayor será la probabilidad). Las fichas que seleccione se bordearán de color azul; sólo se cambiarán las fichas seleccionadas que estén sobre el atril, no las que estén sobre el tablero.

Pasar: Si el jugador tiene un palabrón pero no puede encajarlo con ninguna palabra del tablero y desea esperar o, por cualquier otro motivo, con pulsar este botón cederá su turno.

2.4.5. Jugadores de la Mesa

Debajo de los tres botones principales, aparece una tabla con los nombres de los jugadores de la mesa, y su puntuación. Esta tabla se irá actualizando después de cada jugada, con las puntuaciones obtenidas por cada uno de los jugadores. Estáte atento a la tabla para controlar quién va ganando!

2.4.6. Pausar

Si el jugador necesita hacer una breve pausa durante la partida, pulsará este botón. Si es Jefe de Mesa, la partida se pausará directamente durante un minuto; si no, el jefe de Mesa será quien decida si aceptar o no la pausa. Cada jugador puede solicitar hasta tres pausas en una partida.

2.4.7. Abandonar

Con pulsar este botón, el jugador volverá a la pantalla de la Sala. Si la partida no estaba en curso, no habrá ningún problema. Sin embargo, si se estaba disputando una partida, se le penalizará con 50 puntos negativos en tu puntuación global; también se incrementará en una unidad el número de partidas abandonadas en su historial de juego. Es importante no tomar la costumbre de abandonar partidas porque aparecerá reflejado en los datos de usuario del jugador: nadie quiere jugar con tramposos. (Nota: la penalización y el número de partidas abandonadas sólo se modificarán si el jugador es un usuario registrado y la partida se disputaba en modo Competición).

2.4.8. Chat

Como en el Salón Principal y en la Sala, se dispondrá de un chat con el que el jugador podrá comunicarse con el resto de jugadores. Si la partida aún no ha comenzado, podrá comunicarse también con los jugadores que están en su misma sala; una vez comience la partida, sólo podrá comunicarse con los jugadores de la mesa. Además, el chat aquí tendrá una función informativa: indicará los cambios de turno, las acciones de un jugador y, en el caso de una jugada incorrecta, la/s palabra/s que no existen.

2.4.9. Tiempo

Cuando le toque el turno, cada jugador dispondrá de un tiempo máximo en el que realizar su jugada: puede tratarse de 30, 45 segundos o un minuto, dependiendo de la configuración de juego elegida. El tiempo de turno restante aparecerá debajo del tablero.

2.4.10. Puntuación

La puntuación acumulada de la partida aparecerá en la esquina inferior derecha.

2.4.11. Configuración del juego

Como se ha comentado, la configuración del juego sólo podrá ser elegida por el Jefe de Mesa, y antes de que la mesa se llene. Una vez hayan entrado todos los jugadores a la mesa, se les mostrará una pantalla con la configuración, que podrán aceptar o no.

El Jefe de Mesa, para cambiar la configuración, debe pulsar el botón que está al lado del panel de tiempo. Además de elegir el modo de juego (150, 250 puntos o "Acabar Fichas") y el tiempo máximo de cada turno (30, 45 o 60 segundos), podrá indicar si desea permitir que jueguen o no invitados en la mesa.

2.4.12. Turnos

La partida, desde que comienza hasta que termina, se regulará por turnos. En la primera partida, el primer turno le corresponderá al Jefe de Mesa. Los siguientes turnos se realizarán en el orden en que cada jugador entró en la mesa. Una vez finalizada la primera partida, el primer turno corresponderá al jugador que menor puntuación haya sacado en la partida anterior, siguiendo el mismo orden de rotación de turnos que se utilizó anteriormente.

2.4.13. Modo Competición

Las modificaciones en las puntuaciones y los ELO, así como las estadísticas de partidas (partidas jugadas, ganadas, perdidas, abandonadas...) sólo se modificarán si la partida se disputa en modo Competición, esto es, si todos los jugadores que disputan la partida son usuarios registrados. Si al menos un jugador es un invitado, la partida no se disputará en modo Competición.

2.4.14. Fin de la partida

La partida terminará dependiendo de la configuración de juego elegida. Si se ha elegido el modo 150 puntos o el modo 250 puntos, la partida finalizará cuando un jugador alcance o supere dicha puntuación. En el caso de elegir el modo Acabar fichas, la partida terminará cuando el saco se quede sin fichas. Una vez terminada la partida, se le dará la opción al jugador de comenzar o no otra.

2.4.15. La Ficha Comodín

Como se ha comentado, en QUIJIBO hay dos fichas en blanco, cuyo valor es cero, que pueden ser sustituidas por cualquier letra.

Si un jugador tiene una ficha comodín y desea utilizarla, sólo tiene que moverla al tablero y clicar sobre ella con el botón derecho. Se le abrirá una ventana con todas las opciones posibles; el jugador deberá elegir la letra que desee y la ficha cambiará automáticamente. Es importante asegurarse de que es la letra que quiere, ya que, una vez se ha elegido, no habrá vuelta atrás.

2.5. Cálculo de ELO

¿Qué es el ELO y cómo se calcula?

El ELO es un valor numérico que se asigna a cada jugador en relación con su "fuerza" o nivel de juego, en base al cual se puede estimar su probabilidad de victoria en el futuro. Cuanto mejor es un jugador, mayor es su ELO:

La forma en que se determina en ELO de un jugador fue ideada para el ajedrez en 1959, por el profesor de matemáticas, el Dr. Éló Árpád.

La aportación del profesor Éló fue la de establecer una función que a partir de la diferencia entre los puntos obtenidos por ambos jugadores (después de enfrentarse un número suficiente de veces) proporciona la diferencia de ELO entre ellos.

Este sistema transforma la puntuación obtenida en un torneo (partida ganada = 1 punto, pérdida = 0 puntos, empate = 0.5), en un conjunto de valores numéricos discretos de cuatro cifras enteras que se denomina escalafón.

El escalafón proporciona información estadística confiable sobre el rendimiento de un jugador en el tiempo. Su fundamento teórico se basa en que la diferencia de escalafón entre dos jugadores constituye una guía para predecir el resultado de una partida entre ellos.

Las fórmulas de transformación corresponden al dominio de la teoría de probabilidades y estadística. Si bien es cierto que es imposible obtener una medición exacta de la fuerza de un jugador, la experiencia demuestra que los valores de escalafón, considerados como promedios de rendimiento dentro de un rango específico, se corresponden con la realidad de los resultados cotidianos de un jugador.

Como se ha comentado anteriormente, el sistema ELO se ideó originariamente para el ajedrez. Debido a la fuerte incidencia del factor azar en QUIJIBO (a diferencia del ajedrez, donde la suerte no influye como factor en el juego), la adopción de los parámetros fijados para el ajedrez en QUIJIBO no es del todo conveniente. Examinando los datos de los que se dispone, no costaría demasiado corroborar que la probabilidad de victoria en función de la diferencia de ELO para el juego de QUIJIBO está claramente sobreestimada, la suerte puede hacer (y lo hace con frecuencia) que un jugador venza a otro que está 500 puntos ELO por encima de su escalafón.

Es por ello que se decidió adoptar un sistema basado en el uso de un mismo ELO inicial igual para todos los jugadores (1000), con variación retroactiva causada por la disputa de partidas.

Fórmulas de transformación

Predicción de Ganancia(We)

Es el valor probabilístico esperado para ganar, cuando un jugador se enfrenta a otro, calculado a partir de su diferencia de escalafón con el del oponente.

$$We = \frac{1}{10^{(-dr / 400)} + 1}$$

Ilustración A.13 Predicción de ganancia

Donde,

We = Predicción de Ganancia.

dr = diferencia de escalafón.

En valor We puede variar entre 0 y 1 cuando dr varía desde menos infinito hasta más infinito.

Cálculo de la Variación ELO:

Siendo:

R = rating (escalafón del jugador

n = número de oponentes

Ri = Escalafón ELO de cada uno de los oponentes

1. Calcular D_i según la fórmula:

$$D_i = R - R_i$$

Ilustración A.14 Diferencia de escalafón

Es decir, se calcula la diferencia de escalafón del jugador con respecto a los de los demás contrincantes.

2. Calcular W_{ei} con la fórmula vista anteriormente para cada una de las diferencias de escalafón.
3. Calcular R_v (variación total de escalafón), como el sumatorio de $i=1$ hasta $n-1$ de R_{vi}

Siendo $R_{vi} = K \cdot (P_i - W_{ei})$

donde

R_{vi} = Variación de escalafón del jugador respecto al jugador i

P_i = Puntaje del jugador con respecto al jugador i ($P=1$ si gana jugador; $P=0.5$ si empatan; $P=0$ si pierde jugador)

K = factor de multiplicación ($K=25$ para jugadores con menos de 50 partidas; $K=50$ para jugadores con 50 o más partidas)

¿Hay alguna relación entre el ELO y los puntos de un jugador?

No necesariamente, ya que se obtienen puntos aunque la partida se pierda, con lo que un jugador podría tener una gran cantidad de puntos solamente por jugar más partidas que el resto, aunque hubiera ganado solamente la mitad de ellas.

Los puntos son acumulativos mientras que el ELO es un coeficiente que mide la "calidad" de un jugador. Si quisiéramos dar un valor aproximado de la calidad de un jugador sin usar ELO, podríamos tomar la relación entre el número de partidas ganadas sobre el total de partidas jugadas. Esto aún tendría el inconveniente de que no se sabría si las partidas se ganaron contra novatos o expertos. Con el ELO este dato también queda contemplado.

¿Qué ELO tiene un invitado?

Los invitados, jugadores no registrados, no tienen ELO asignado al no disponer de un historial anterior. Esto tiene una consecuencia importante, y es que si en una partida de cualquier número de jugadores hay al menos un invitado, no se modifica el ELO de ningún jugador.

Los jugadores que deseen subir en el ranking ELO deberán enfrentarse a otros jugadores registrados.

¿Cuándo se modifica el ELO?

El ELO se modifica al finalizar una partida con victoria (o empate), y siempre que no haya invitados jugando.

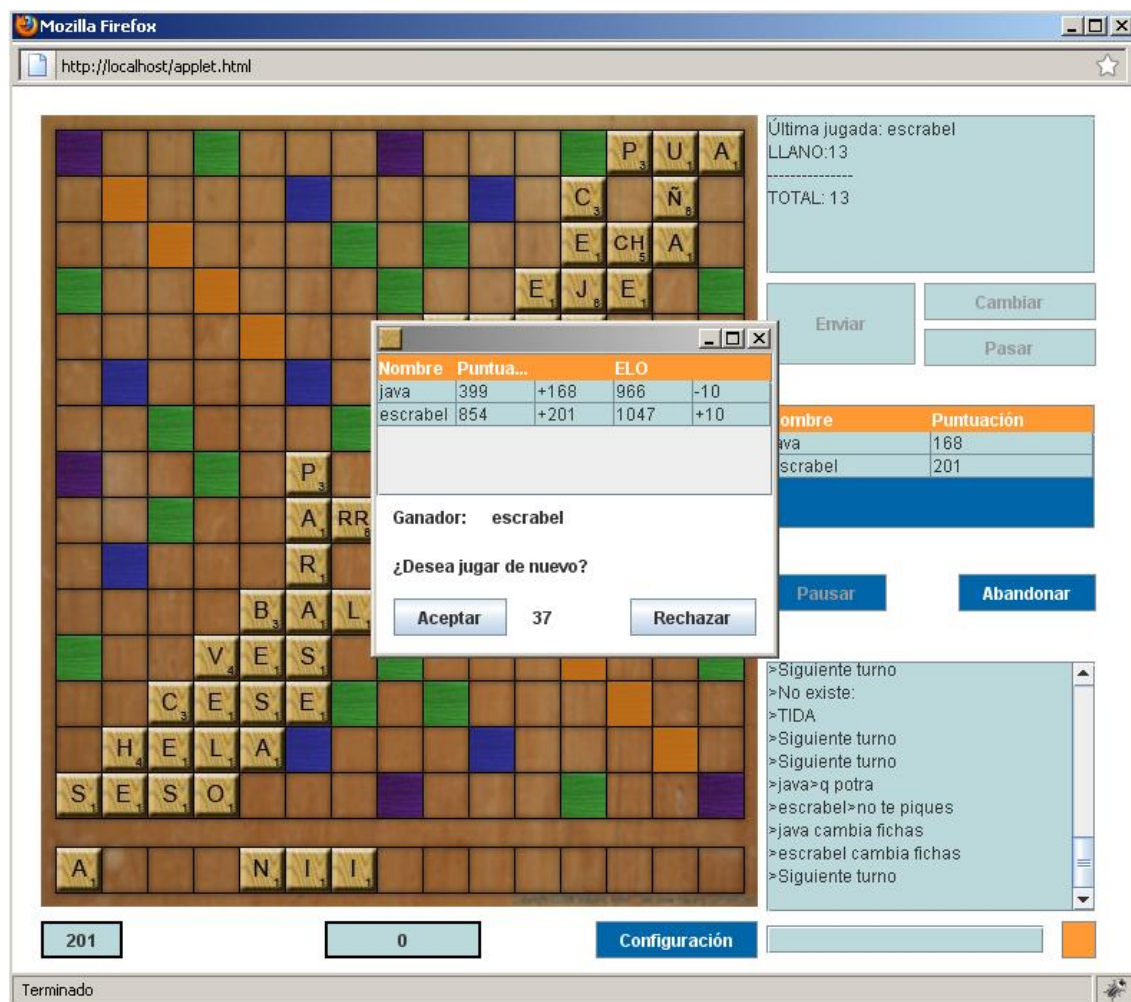


Ilustración A.15 Final de Partida

DESARROLLO DE UN SISTEMA MULTIJUGADOR EN RED SOBRE UN JUEGO BASADO EN EL “SCRABBLE”

Alumno: Carolina Flores Ugarrita

Titulación: Ingeniería Técnica en Informática de Gestión

Índice

- Introducción
- Desarrollo
- Líneas futuras
- Conclusiones
- Demostración
- Tiempo de preguntas

Introducción

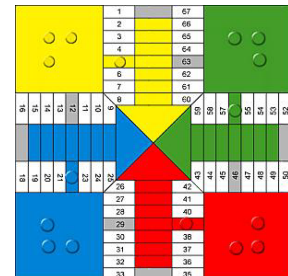
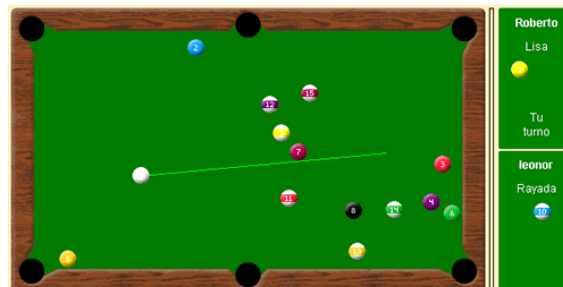
- Motivación
- Juegos en red multijugador
- En qué consiste el Scrabble
- QUIJIBO
- Tecnologías empleadas

Motivación

- Motivación Personal
- Desarrollo de un Sistema complejo que pone en práctica casi todos los aspectos vistos en la carrera.

Juegos en red multijugador

- Gran popularidad en los últimos años.
- Juegos de todo tipo.
- Posibilidad de competir con cualquier persona, de una manera rápida y sencilla.



En qué consiste el Scrabble

- De 2 a 4 jugadores.
- Tablero 15x15 con casillas de distintos valores.
- 100 fichas con diferentes letras y valores que se van repartiendo a los jugadores.
- Objetivo: Formar palabras que existan en el diccionario obteniendo la mayor puntuación posible.

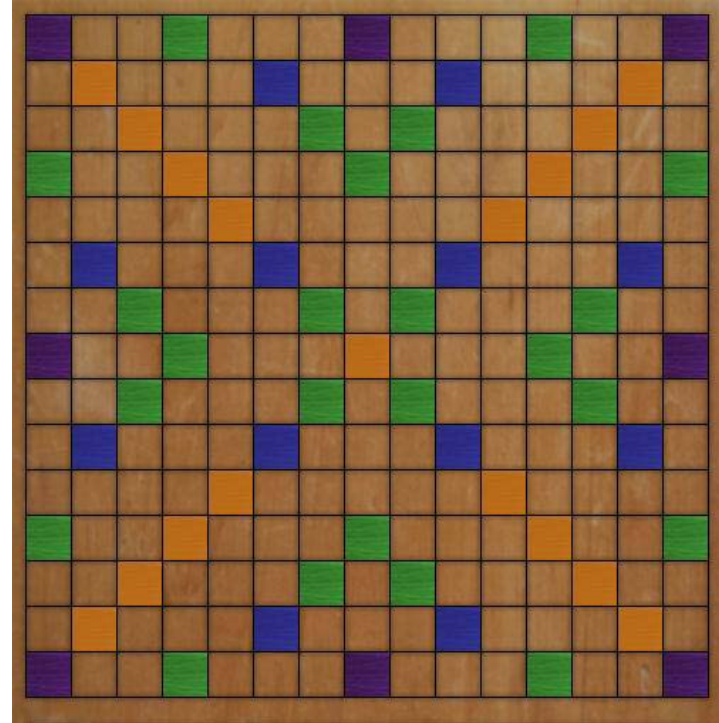
En qué consiste el Scrabble

Casilla Naranja: x2 Palabra

Casilla Morada: x3 Palabra

Casilla Verde : x2 Letra

Casilla Azul: x3 Letra



En qué consiste el Scrabble

- Disposiciones generales:
 - Las palabras están formadas por dos o más letras.
 - Colocación en el tablero: horizontal o verticalmente.
 - Palabras leídas de izquierda a derecha o de arriba hacia abajo.
 - La jugada inicial debe ocupar la casilla central.
 - Cada palabra sobre el tablero debe empalmarse con alguna de las incorporadas.

En qué consiste el Scrabble

- Desarrollo de la Partida:
 - Al comenzar la partida, se reparte a cada jugador siete fichas del saco.



- Las dos fichas en blanco o comodines pueden utilizarse en sustitución de cualquier letra.



En qué consiste el Scrabble

- Desarrollo de la partida
 - El valor de los comodines es de cero.



- Los dígrafos RR, CH, LL aparecen en una sola ficha. Por tanto, no podrán utilizarse dos eres, ni dos eles, ni la ce y la hache para formar una doble letra.



En qué consiste el Scrabble

- Recuento de puntos:
 - El jugador obtiene puntos por todas las nuevas palabras que forme durante su turno.
 - Valor de cada letra: está indicado con una cifra en la parte inferior de la ficha.
 - Valor de cada palabra : Suma del valor de todas las letras (teniendo en cuenta las casillas con premio).
 - Los premios para las letras y palabras aplican solamente la primera vez que se ocupa la casilla de premio.

En qué consiste el Scrabble

- Recuento de puntos:
 - Cuando en un mismo turno el jugador forma dos o más palabras nuevas, se suma la puntuación de todas ellas. La(s) letra(s) comun(es) entre ellas se cuenta(n) en cada palabra, incluido el premio, si lo tiene(n).
 - Palabrón: Cuando un jugador coloca las 7 fichas de su atril en un solo turno, obtiene 50 puntos extra como premio.

En qué consiste el Scrabble

- Cambio de Fichas:
 - Un jugador puede utilizar su turno para cambiar algunas fichas de su atril, o todas si así lo desea. El jugador que ha cambiado fichas debe esperar al siguiente turno para jugar.
 - Está permitido pasar en cualquier momento del juego, sin cambiar fichas.

En qué consiste el Scrabble

- Control del tiempo:
 - La partida se desarrollará por medio de turnos ordenados con una determinada duración máxima.



- La partida terminará dependiendo del modo de juego elegido: bien cuando se alcance una determinada puntuación, o bien cuando se haya repartido la totalidad de las fichas.

QUIJIBO

- Objeto del proyecto: realización de un Sistema completo que permita a los usuarios poder jugar entre ellos en tiempo real.
- El acceso al juego se realizará a través de un página Web en cuyo contexto se ejecutará la aplicación.
- Además de acceder al juego, el usuario podrá registrarse, acceder a su cuenta, consultar Rankings, buscar jugadores...

QUIJIBO

- Juego diseñado para jugar para pasar el rato, o para competir con otros jugadores.
- Dos tipos de jugadores: invitados (anónimos) y registrados (se almacenan sus datos y estadísticas de juego).

Tecnologías empleadas

- Lenguaje de Programación:
 - Para la parte WEB: HTML, PHP y JavaScript.
 - Para la realización de la aplicación del juego (la parte del cliente y del Servidor): JAVA2SE

Tecnologías empleadas

- ¿Por qué JAVA?
 - La programación orientada a objetos.
 - La comodidad que supone trabajar con sockets y streams para enviar y recibir datos en red.
 - La necesidad de utilizar hilos para las conexiones cliente-servidor.
 - La facilidad para crear interfaces gráficas de usuario.
 - Poder trabajar con bases de datos.

Tecnologías empleadas

- ¿Por qué JAVA?
 - La compatibilidad con diferentes plataformas.
 - De especial interés se trata la utilización de Applets.



Tecnologías empleadas

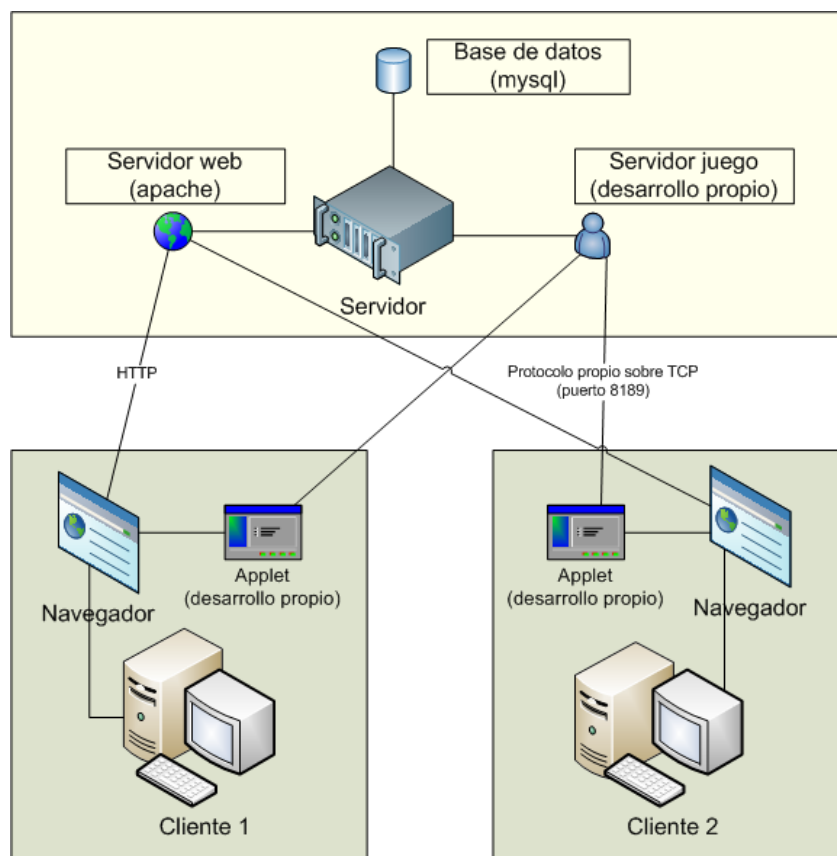
- Applets:
 - Su empleo de se trata de uno de los principales objetivos del proyecto.
 - Comodidad para el usuario: la aplicación no requiere instalación, gracias a la utilización de Applets de Java.

Desarrollo

- Arquitectura
- Cliente
 - Interfaz de usuario
 - Clases
- Servidor
- Protocolo
 - Clases Protocolo, Enviador y Recibidor
 - Paquetes

Arquitectura

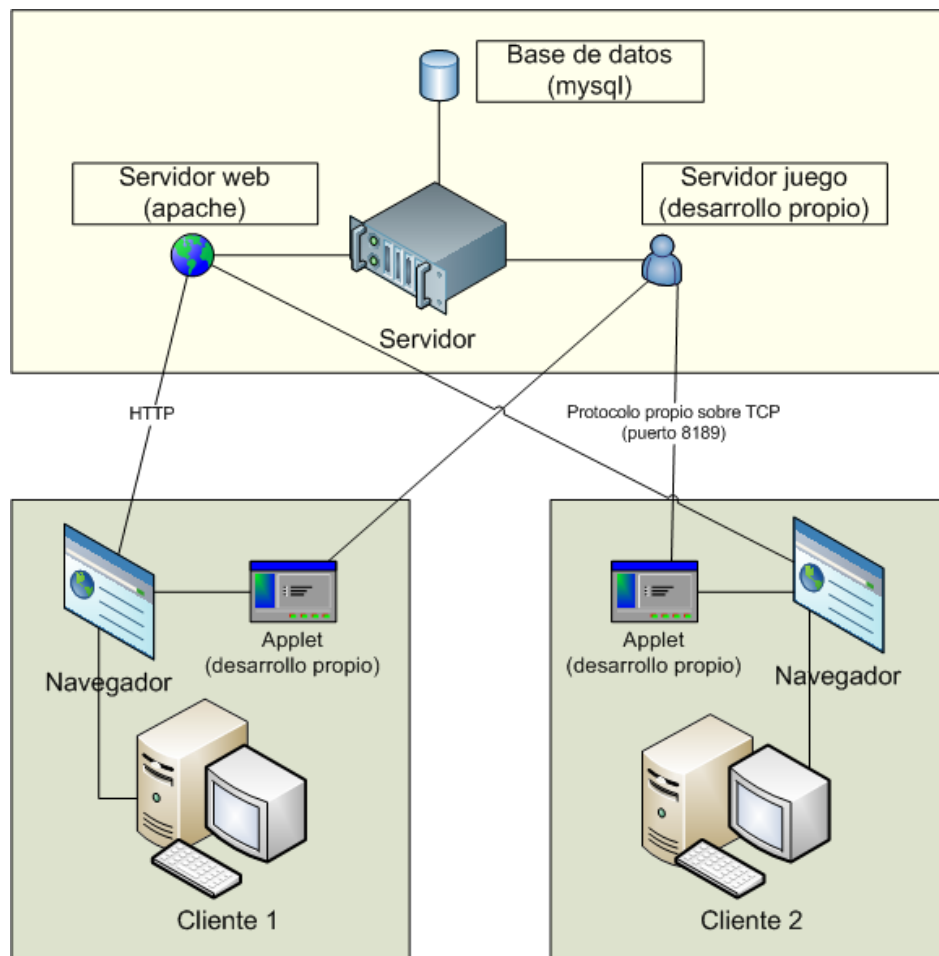
- La arquitectura del Sistema es la siguiente:



Arquitectura

- Servidor: En el Servidor se ejecutarán el Servidor Web Apache, la base de datos y el Servidor del juego en sí, de desarrollo propio.
- Cliente: El Cliente realiza la petición de la página al Servidor Web, en cuyo contexto podrá ejecutar el Applet y realizar la petición de conexión al Servidor del juego.

Arquitectura



Cliente

- Parte que se ejecutará en el Applet del Cliente.

Cliente

- Interfaz de Usuario:
 - Pantalla de Login
 - Salón Principal
 - Sala
 - Mesa

Cliente

- Login:

The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://localhost/applet.html`. The main content area features a blue header with the text "QUIJIBO" in a stylized font. Below this, there are two login panels. The left panel is titled "Invitados" and contains a "Nick" input field and an "Entrar" button. The right panel is titled "Registrados" and contains "Nick" and "Password" input fields, both with example text ("escribel" and "*****" respectively), and an "Entrar" button. The browser window has a status bar at the bottom that says "Terminado".

Cliente

- Salón Principal:

Mozilla Firefox
http://localhost/applet.html



Nombre	J.MAX	Jugadores
Sala 0	30	1
Sala 1	30	3
Sala 2	30	2
Sala 3	30	0
Sala 4	30	0
Sala 5	30	1
Sala 6	30	0
Sala 7	32	0
Sala 8	32	0

Entrar

Salas de 2 jugadores: 0,1,2,3

Salas de 3 jugadores: 4,5,6

Salas de 4 jugadores: 7,8

Nombre	P.Global	ELO
java	399	966
hola	151	988
invit	-	-
ayayaya	-	-
quake88	-	-
mr_t	-	-
escrabel	854	1047
bcn90	-	-
wewewe	-	-
goku_jar	-	-

>escrabel ha entrado al Salón

>escrabel>hola

>bcn90 ha entrado al Salón

>bcn90>wenass

>wewewe ha entrado al Salón

>invit>alguien pa 1 d 4?

>quake88>2 jugadores a 1 minuto

>wewewe>tuuu javi

>bcn90>sperate k ahora se mete ste

>goku_jar ha entrado al Salón

>goku_jar>ya

>goku_jar>vamos pues?

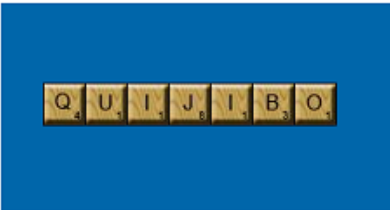
>bcn90>si

Terminado

Cliente

- Sala:

Mozilla Firefox
http://localhost/applet.html



Nombre	J.MAX	Jugadores
Mesa 0	2	1
Mesa 1	2	0
Mesa 2	2	0
Mesa 3	2	0
Mesa 4	2	0
Mesa 5	2	0
Mesa 6	2	0
Mesa 7	2	0
Mesa 8	2	0

Jugadores de la mesa

java

Entrar

Volver

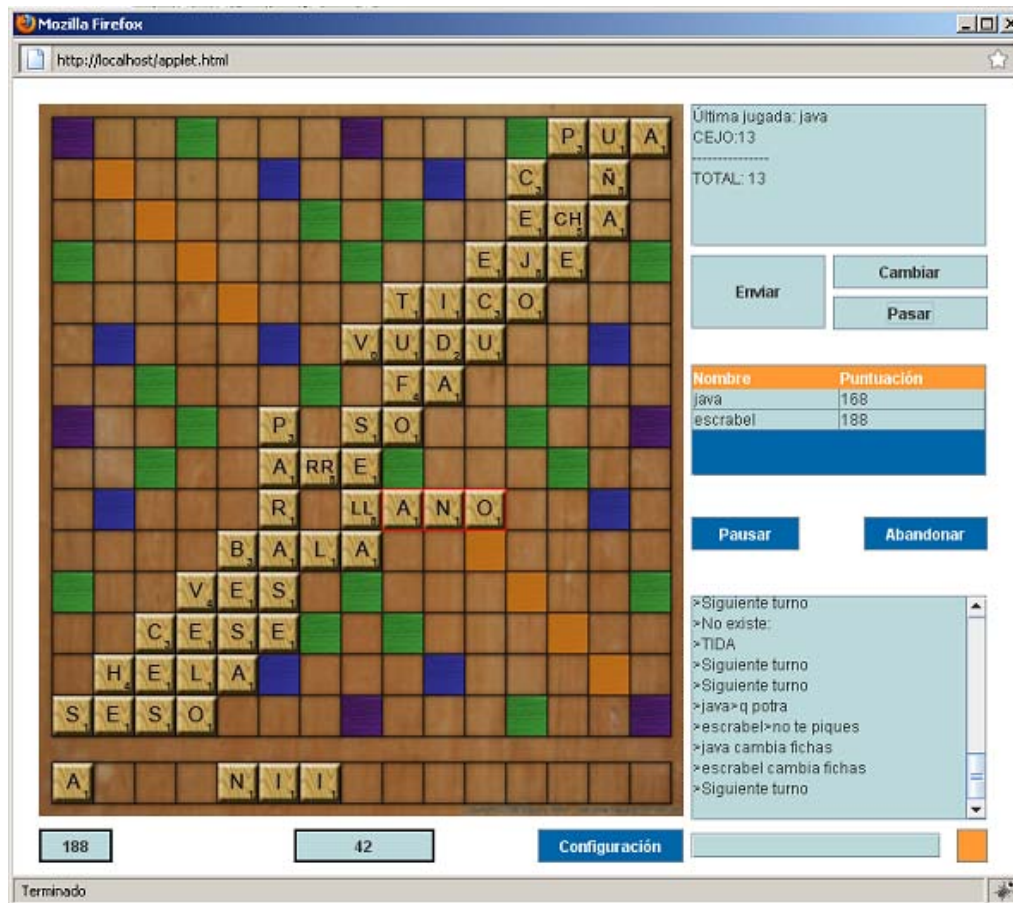
Nombre	P.Global	ELO
wtf1	-	-
wewewe	-	-
goku_jar	-	-
bcn90	-	-
quake88	-	-
hola	151	988
mr_t	-	-
escrabel	854	1047
java	399	966

>escrabel ha entrado en Sala 0
>java ha entrado en Sala 0
>escrabel>a la mesa 1
>bcn90>xDDD
>java>ok
>java ha entrado en la Mesa 0

Terminado

Cliente

- Mesa:



Cliente

- Clases:
 - Listado y breve descripción de las clases del Cliente.

Cliente

- Clases:
 - Controlador: clase desde la que se tiene acceso a todas las demás, para inicializarlas y para que éstas tengan acceso entre ellas).
 - Cliente: se utiliza para crear un socket y realizar una petición de conexión al Servidor, mediante el host y el puerto indicados. También se encarga de cerrar el socket si se cierra la conexión.

Cliente

- Clases:
 - Enviador: Esta clase, que hereda de Thread, se encarga fundamentalmente de enviar paquetes de una cola bloqueante utilizando el Protocolo, que se verá más adelante.
 - Recibidor: Esta clase, que hereda de Thread, se encarga de recibir paquetes por medio del Protocolo y de realizar las acciones asociadas a ellos.

Cliente

- Clases:
 - PanelJuego: Se encarga de realizar el dibujo del tablero y del atril y de procesar correctamente los eventos del ratón para lograr emular la funcionalidad del juego ajustándose al estado de la partida.
 - JugadorInfo: Contiene información que le llega del Servidor sobre el cliente como jugador: el nombre, la puntuación general y de la partida, ELO...

Cliente

- Clases:
 - Sala: Contiene información que le llega del Servidor sobre una determinada sala: el nombre de la sala, la capacidad máxima, la lista de jugadores de la sala y la lista de las mesas...
 - MesaInfo: Contiene información que le llega del Servidor sobre una determinada mesa: el nombre de la mesa y de la sala en la que se encuentra, el número de jugadores necesarios y una lista de jugadores que se encuentren en la mesa.

Cliente

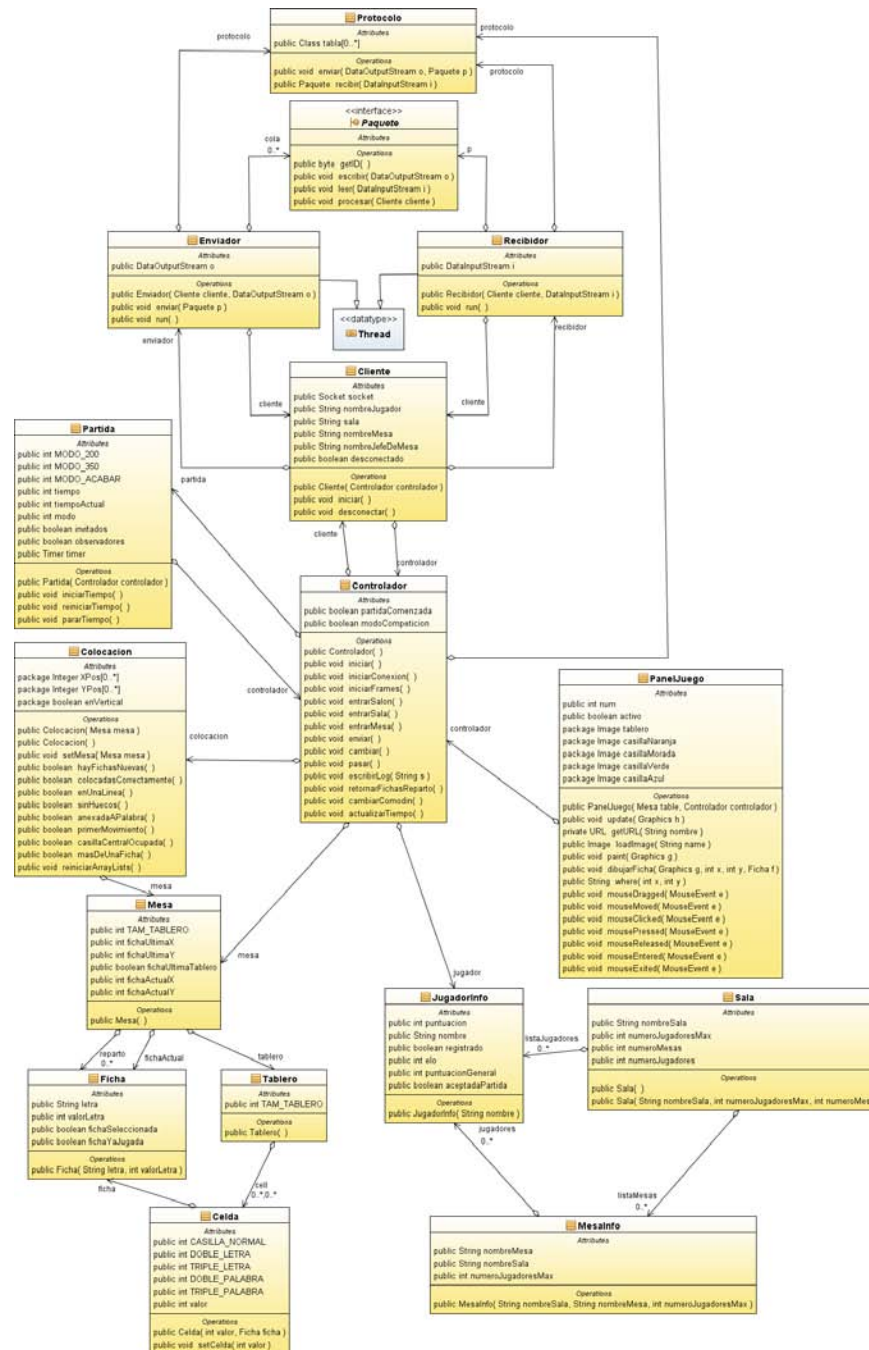
- Clases:
 - Partida: Se utiliza para que al jugador se le muestre un contador del tiempo de cada turno.
 - Mesa: Esta clase consta de un tablero y un reparto, que será lo que el usuario verá cuando esté en la mesa. Se irá actualizando con los paquetes que le lleguen del Servidor.

Cliente

- Clases:
 - Tablero: Esta clase consta de una matriz de celdas 15x15, asignándole a cada celda un determinado valor según su disposición en el tablero.
 - Celda: Esta clase tiene el propósito de simular las casillas del tablero y del reparto: una celda podrá tener un determinado valor y podrá estar ocupada o no por una determinada ficha.

Cliente

- Clases:
 - Ficha: Esta clase se encarga de simular las fichas del juego: una ficha tendrá una determinada letra asignada y un valor. Además, una ficha podrá o no haber sido jugada o tratarse de una ficha que se encuentra seleccionada.
 - Colocación: Esta clase se encarga de comprobar si las fichas sobre el tablero están correctamente colocadas .



Servidor

- El Servidor del Juego.
- Clases:
 - Listado y breve descripción de las clases del Cliente.

Servidor

- Clases:
 - Servidor: Se encarga de:
 - Escuchar y aceptar las peticiones de conexión.
 - Mantener la lista de Clientes del Sistema.
 - Mantener la lista de Jugadores del Salón Principal.
 - Mantener la lista de Salas.
 - Cargar el diccionario.

Servidor

- Clases:
 - ClienteHandler: Se encarga de:
 - Crear un dataInputStream que usa un input stream para leer bytes del socket.
 - Crear un dataOutputStream que usa un output stream para escribir bytes en el socket.
 - Crear e iniciar la ejecución de dos hilos por cada cliente que se encargarán de enviar y recibir paquetes.
 - Realizar las acciones oportunas cuando se produzca la desconexión de un cliente.

Servidor

- Clases:
 - Enviador: Esta clase, que hereda de Thread, se encarga fundamentalmente de enviar paquetes de una cola bloqueante utilizando el Protocolo.
 - Recibidor: Esta clase, que hereda de Thread, se encarga de recibir paquetes por medio del Protocolo y de realizar las acciones asociadas a ellos.

Servidor

- Clases:
 - Diccionario: Esta clase se encargará de meter todas las palabras de un fichero en una hashtable, para después poder comprobar si una determinada palabra es una clave de la hashtable.
 - JugadorInfo: Esta clase se encargará de mantener información general del cliente como jugador, como su nick, si es usuario registrado o no, su puntuación global, su ELO...

Servidor

- Clases:

- Jugador: Esta clase guarda una estrecha relación con JugadorInfo; se encarga de mantener la información de un jugador en una partida (la puntuación de la partida, las fichas que se le asignan...)

Además consta de determinados métodos para, en el caso de tratarse de un jugador registrado, consultar y actualizar sus datos en la base de datos .

Servidor

- Clases:
 - Sala: Se encarga de mantener una lista de jugadores de la sala y la lista de mesas.
 - MesaInfo: Se encarga fundamentalmente de mantener información sobre la mesa, nombre, el número de jugadores requeridos, si ha comenzado una partida, la lista de jugadores de la mesa, el Jefe de Mesa, el modo de juego y tiempo por turno, si la partida se disputa en modo Competición, si se permiten invitados...

Servidor

- Clases:
 - Mesa: Esta clase guarda una estrecha relación con MesaInfo. Constará de determinados métodos que realizarán las acciones necesarias para el desarrollo del juego (Repartir fichas aleatoriamente; comprobar una jugada y, si es correcta, fijar las fichas y restituir las fichas del jugador; intercambiar fichas; cambiar de turno, etc).

Servidor

- Clases:
 - Partida: Esta clase se encarga de gestionar el tiempo de la partida, los turnos y su duración, así como de las posibles pausas que puedan ser solicitadas.
 - Saco: Esta clase consta de una lista de las fichas por repartir. Esta lista contiene todas las fichas del juego al inicio de la partida y conforme éstas se vayan repartiendo, se eliminarán de la lista.

Servidor

- Clases:
 - Tablero
 - Celda
 - Ficha
 - Colocación

Servidor

- Clases:
 - Palabras: Esta clase se encarga de:
 - Obtener la/s nueva/s palabra/s formada/s.
 - Calcular la puntuación obtenida por cada una de ellas, atendiendo al valor de las fichas y de las casillas del tablero.
 - Comprobar si una jugada ha sido “Palabrón”.
 - Calcular el total de puntos obtenidos en la jugada.



Protocolo


- Para la comunicación entre el Servidor y el Cliente, se ha elaborado un Protocolo propio sobre TCP, basado en el envío y recepción de paquetes.
- Estos paquetes son instancias de clases que implementan el interfaz Paquete.

Protocolo

- En el Servidor y en el Cliente se crearán dos hilos por cada Cliente: Enviador y Recibidor. Estarán continuamente ejecutándose hasta que la conexión se interrumpa.
- Enviador: utiliza el Protocolo para ir enviando los paquetes (de una cola bloqueante).
- Recibidor :se encargará de la recepción de paquetes por medio del Protocolo, y de realizar las acciones asociadas a ellos.

Protocolo

- La Clase Protocolo:

 Protocolo
<i>Attributes</i> <u>public Class tabla[0..*]</u>
<i>Operations</i> public void enviar(DataOutputStream o, Paquete p, String cliente) public Paquete recibir(DataInputStream i, String cliente)

Protocolo

– Atributos:


- tabla: Tabla de elementos que representan las clases de cada paquete.

– Métodos:

- enviar(): El Enviador llamará a este método cada vez que se quiera enviar un paquete (serialización).
- recibir(): El Recibidor llamará a este método a la espera de la recepción de paquetes. Utilizando la Tabla, se comprobará qué tipo de Paquete es, y se creará una instancia del mismo a la que se le rellenarán los atributos (deserialización).

Protocolo

- El interfaz Paquete

<<interface>>
 Paquete
<i>Attributes</i>
<i>Operations</i>
<i>public byte getID()</i>
<i>public void escribir(DataOutputStream o)</i>
<i>public void leer(DataInputStream i)</i>
<i>public void procesar(ClienteHandler cliente)</i>

Protocolo

- El interfaz Paquete

- Métodos:

- getId(): devolverá un número que identificará de manera única a cada paquete.
 - escribir(): será llamado por el Protocolo para escribir el paquete en el stream de salida del socket.
 - leer(): será llamado por el Protocolo para, una vez creada una instancia del paquete recibido, leer el paquete del stream de entrada del socket.
 - procesar(): se indicará qué se debe hacer cuando se reciba un determinado paquete.

Protocolo

- A continuación se mostrarán los diferentes tipos de paquetes que implementan el interfaz Paquete.

Protocolo

- PaqueteError:
 - Del Servidor al Cliente, cuando se produzca cualquier tipo de situación anómala.
 - Contiene el tipo de error, el mensaje a mostrar y un booleano que indicará si es fatal o no(en cuyo caso el cliente desconectará).

Protocolo

- PaqueteLogin:
 - Del Cliente al Servidor para identificarse con sus credenciales.

Protocolo

- PaqueteSalon:
 - Del Servidor al Cliente, para mostrar al Cliente la información del Salón Principal: la lista de Salas y de jugadores del Salón.

Protocolo

- PaqueteSelecSala:
 - Del Cliente al Servidor, cuando el cliente seleccione una sala para acceder a ella.

Protocolo

- PaqueteSala:
 - Del Servidor al Cliente, para mostrar al Cliente la información de la Sala seleccionada : la lista de Mesas y de jugadores de la Sala.

Protocolo

- PaqueteChatSalon:
 - Del Cliente al Servidor con su nick y el mensaje enviado.
 - Del Servidor se reenviará a todos los clientes del Salón.

Protocolo

- PaqueteChatSala:
 - Del Cliente al Servidor con su nick y el mensaje enviado.
 - Del Servidor se reenviará a todos los clientes de la misma Sala.

Protocolo

- PaqueteJugadorEntraSalon:
 - Del Servidor a todos los clientes del Salón, cada vez que un cliente entre en el Salón Principal, para actualizar la información.

Protocolo

- PaqueteJugadorEntraSala:
 - Del Servidor a todos los clientes de la Sala, cada vez que un cliente entre en dicha Sala, para actualizar la información.

Protocolo

- PaqueteVolverSalon
 - Del Cliente al Servidor, cada vez que desee volver al Salón Principal desde una Sala.

Protocolo

- PaqueteSaleSala:
 - Del Servidor a todos los clientes de una Sala, cuando un Cliente ha regresado al Salón Principal, para actualizar dicha información.

Protocolo

- PaqueteJugadorSeVa:
 - Del Servidor a los Clientes para informar de la desconexión de un determinado Cliente.

Protocolo

- PaqueteSelecMesa:
 - De Cliente al Servidor, cuando se seleccione una Mesa para acceder a ella.

Protocolo

- Paquete JugadorEntraMesa:
 - Del Servidor a los Clientes de la Sala para informar cuando un jugador entre en una mesa.

Protocolo

- PaqueteMesa:
 - Del Servidor al Cliente para mostrarle la información de la Mesa cuando accede a ella: la lista de jugadores de la Sala, el nombre del Jefe de Mesa y el estado de la mesa.

Protocolo

- PaqueteConfig:
 - Del Cliente al Servidor, con el propósito de cambiar la configuración de la Mesa.
 - En el Servidor se comprobará si el paquete lo ha enviado el Jefe de Mesa antes del comienzo de la partida; en ese caso, se aplicarán los cambios en la configuración de la partida.

Protocolo

- PaqueteSalirMesa:
 - Del Cliente al Servidor, cada vez que el Cliente desee abandonar la Mesa en la que se encuentre.
 - En el caso de estar una partida en curso, se realizarán las acciones convenientes.

Protocolo

- PaqueteSaleMesa:
 - Del Servidor a todos los Clientes de la Sala para informar de que un Cliente ha abandonado una mesa.

Protocolo

- PaqueteChatMesa:
 - Del Cliente al Servidor con el nick del Cliente y el mensaje enviado.
 - Del Servidor a los Clientes de la mesa si la partida ha comenzado y si no, a todos los Clientes de la Sala.

Protocolo

- PaqueteActualizacion:
 - Del Servidor a todos los Clientes de la Sala para actualizar la información sobre los ELO y Puntuaciones Globales de los jugadores una vez acabada la partida de una mesa.

Protocolo

- PaqueteAceptarConfig:
 - Del Servidor a los Clientes que se encuentren en una misma mesa antes de comenzar una partida, con la configuración del juego.

Protocolo

- PaqueteConfigAceptada:
 - Del Cliente al Servidor para indicar que se acepta la configuración del juego.
 - En el Servidor se llevará una cuenta de los jugadores que aceptan la configuración. Si antes de agotarse el timeout todos los jugadores han aceptado, se procederá al comienzo de la partida.

Protocolo

- PaqueteComenzarPartida:
 - Del Servidor a los Clientes de la mesa para comenzar la partida.
 - El Servidor enviará este paquete con los datos de la configuración, el modo de juego y el reparto inicial correspondiente, a cada uno de los jugadores de la mesa.

Protocolo

- PaquetePausarJuego:
 - La pausa sólo podrá ser accionada a través del Jefe de Mesa. Este paquete se enviará:
 - Si lo envía el Jefe de Mesa, del Cliente al Servidor, y después del Servidor a todos los Clientes de la Mesa.
 - Si lo envía un jugador normal, del Cliente al Servidor, del Servidor al Cliente que sea Jefe de Mesa; y si éste acepta la solicitud de pausa, de nuevo al Servidor.

Protocolo

- PaqueteReanudarJuego:
 - Del Servidor a los Clientes de la Mesa, para informarles del fin de la pausa y de la reanudación del juego.
 - Se desbloquearán el tablero, el reparto y los botones de juego del jugador al que corresponde el turno actual.

Protocolo

- PaqueteJugadorMueveFicha:
 - Del Cliente al Servidor cada vez que un jugador mueve una ficha.
 - Del Servidor a los demás Clientes de la Mesa, para informarles del movimiento.

Protocolo

- PaqueteTurno:
 - Del Servidor todos los Clientes de la mesa cada vez que se produce un cambio de turno.
 - Al recibir este paquete el Cliente, se comprobará si el turno le corresponde y, en este caso, se desbloquearán el tablero, el reparto y los botones del juego.

Protocolo

- PaqueteJugadorEnvia:
 - Del Cliente al Servidor, cada vez que se deseen enviar una jugada y las fichas estén correctamente colocadas sobre el tablero.

Protocolo

- PaqueteJugarFichas:
 - Del Servidor al Cliente, para informar al Cliente del estado del tablero, y del reparto si fuera necesario

Protocolo

- PaqueteInfoJugada:
 - Del Servidor a todos los Clientes de la mesa con la información de una determinada jugada (si ha sido correcta, palabras erróneas, puntuaciones por palabra, puntuación total...).

Protocolo

- PaqueteCambiar:
 - Este paquete se envía:
 - Del Cliente al Servidor, cuando el Cliente desea cambiar las fichas seleccionadas de su reparto.
 - Del Servidor al Cliente, tras haber realizado una jugada correcta para enviarle su nuevo reparto, o cuando el Cliente ha solicitado cambiar su reparto.

Protocolo

- PaquetePasar:
 - Del Cliente al Servidor, cada vez que el jugador desee pasar de turno.
 - Tras recibir el paquete, se procederá al cambio de turno y se enviará a todos los clientes de la mesa un paquete PaqueteTurno.

Protocolo

- PaqueteCambiaComodin:
 - Del Cliente al Servidor cuando desee utilizar la ficha comodín.
 - Del Servidor a los Clientes de la mesa, para informarles del cambio.

Protocolo

- PaqueteAcabarPartida:
 - Del Servidor a los Clientes de la mesa cuando finalice la partida, para informarles del resultado de la misma.
 - Se enviará el nombre del jugador ganador, las puntuaciones, y las variaciones de ELO si la partida se ha disputado en Modo Competición.

Líneas futuras

- Mejoras:
 - Creación de Administradores que puedan expulsar a jugadores del sistema.
 - Añadir la opción de que el Jefe de Mesa pueda expulsar a jugadores de la mesa.
 - Espectadores de una partida.

Líneas futuras

- Mejoras:
 - Timeout en la partida para que tras la desconexión accidental de un usuario (por ejemplo, se pierde la conexión a internet) se pueda reanudar la partida donde se había dejado.
 - En la parte de la Web, se podrían añadir más opciones: (creación de un foro, sistema de registro de validación mediante correo, almacenamiento de más datos de usuario, avatares..)

Conclusiones

- El desarrollo del proyecto ha contribuido notablemente a ampliar la formación adquirida en la carrera.
- Se ha profundizado ampliamente en el conocimiento y uso del lenguaje de programación JAVA.

Demostración

- A continuación, se mostrará un vídeo que permite observar el desarrollo del juego.



Tiempo de Preguntas

